NEDA Annual

North East Digital Association

Devoted to Packet Networking in the North East

Volume 3 **Revised March 27, 1992**

Welcome to the Annual

This publication is shipped to NEDA members annually. All articles from NEDA Quarterlies which still apply to packet radio today are published in the Annual. The Annual also includes documentation needed to construct and operate the NEDA recommended network node equipment.

In the past several years a packet radio network implementation has been very successful in the north east portion of the North American continent. Over seventy sites in at least two provinces and seven states are tied together via dedicated point to point amateur radio links. None of that area had general purpose backbone support three years ago. Excellent progress is being made but there is still a long way to go.

The purpose of the Annual includes being a compendium of club progress, a directory to club activities, a guide to network implementation and an instruction manual for packet network operation. Some forms of packet operation and network building are not represented in this document, yet. In order to promote packet networking NEDA hopes to put all of the available documentation needed to build a successful network into the hands of any who might make good use of it towards the cause. Hopefully this has finally put an end to knowledge hoarding that may have contributed to the lack of progress in packet radio in the eighties.

This document will be revised after each NEDA Quarterly is produced. Articles submitted for the Quarterly that concern the above-mentioned purpose of the Annual will be published here. If you have any comments about this document or any input that should go into the Annual or Quarterly please contact your editor at:

Dana Jonas 518-784-5482
RD. 2 Box 92 WA2WNI @ WA2PVV.NY
Valatie NY 12184
—Editor

Table of Contents

North East Digital Association

The North East Digital Association was formed in 1989 to support packet networking in the north east. The association's main purpose is to support a packet network in the North Eastern region of the United States and in provinces of Canada adjoining. The club supports the network with technical assistance, documentation, promotion and with actual custom hardware when necessary.

NEDA is a member supported organization. Members are those hams who feel that the cause of packet radio networking is worth funding.

NEDA is governed by an elected board of directors. The NEDA Constitution is printed in this document.

The goals of NEDA as specified in the original charter are:

- To form and maintain a reliable and consistent long distance packet network;
- To educate amateur radio operators as to effective methods of long distance packet network construction and operation;
- To provide a common network that can handle the communications demands placed upon it by a wide variety of different users;
- To encourage participation by individuals who could provide equipment for expansion of the network and creation of redundancy to the network's major links;
- To provide a forum for sharing resources.

History

In 1989 Kevin WA2VAM, Dana WA2WNI, Rob KC3BQ, Bob NQ1C, Jim K1MEA, Herb WA1TPP and Tadd KA2DEW met at WA1TPP's house to discuss how to perpetuate a packet network that they had only recently put together which spanned from Boston to Albany NY and was just about to spread to Syracuse. The NEDA name was coined by NQ1C after the group started pondering the idea of a club. The major reason they wanted a club was so that the process of packet networking would persist even if one or more of the founders were no longer involved. No individual should be important enough that the loss of that individual would cripple the network.

As of this writing the club has been in existence for over two years. There have been two annual elections. The Constitution has been modified five times since it's adoption. The club is growing and continues to change in its goals and it's methods of operating. There is still lots of room to change and lots of growing up to do. Most of the founders of the club are still involved in the club but there are many new people since the club's founding which is *excellent*.

The packet network which NEDA members are a part of is now large enough and reliable enough to allow daily round tables between stations that are 500 miles apart. Check out the CROWD node at CANDGA and see for yourself.

Getting Involved

To get involved with NEDA you can approach any of the members or officers via packet or at one of the many flea markets in the north east. A complete membership roster is printed in the Quarterly. You can write to the NEDA PO Box, or you can send mail to the editor. There is a list of Network Regional Coordinators in the Quarterly. The NRCs are volunteers who are committed to promoting packet networking and will work with hams of any level of commitment. They can help with network use, or network expansion. There are now hundreds of hams involved in this project. There is room for thousands. So far as I know there are no VHF terrestrial links from the Atlantic to the Pacific. Why is this? Lets do it!

The following is a list of the hams who were founders or who have served as board members and board appointed officers since the club started.

Herb Belin - WA1TPP
founder
board member 1990
nrs 1990

Kevin Wright - WA2VAM
founder
board member 1990, 1991
nrs 1990, 1991

Dana Jonas - WA2WNI
founder
board member 1990, 1991
alternate 1992
nrs 1990, 1991
secretary 1990, 1991, 1992
editor 1992
co-chair NESAC 1990, 1991, 1992

Rob Marzili - KC3BQ
founder

Jim Wzorek, K1MEA
founder
board member 1990, 1991, 1992
nrs 1991
chairman bbsc 1990, 1991, 1992

Bob Lafleur, NQ1C
founder
board member 1990, 1992
alternate 1991
information manager 1990, 1991, 1992

Tadd Torborg, KA2DEW
founder
board member 1990
alternate 1991
chairman ntech 1990
editor 1990, 1991

Herb Salls, WB1DSW
treasurer 1990, 1991, 1992
membership 1990, 1991, 1992

Cal Calvito, WA1WOK
board member 1992
alternate 1991
co-chair nesac 1990, 1991, 1992

Cal Stiles, W1JFP
board member 1991, 1992
board chair 1991, 1992

Linda Collins, NR1N
board member 1991
alternate 1992
nrs 1990, 1991

Rich Place, WB2JLR
board member 1991, 1992
chairman ntech 1991, 1992
nrs 1991

Howie Cohen, WA2TVE
Chairman HexiPus™ 1991, 1992

Bob Seger, WB2JLR
board member 1992
alternate 1991

Don Russ, N2CZL
alternate 1991

Russ McAllister, WA1TLN
alternate 1991, 1992

Chris Piggot, WZ2B
alternate 1992

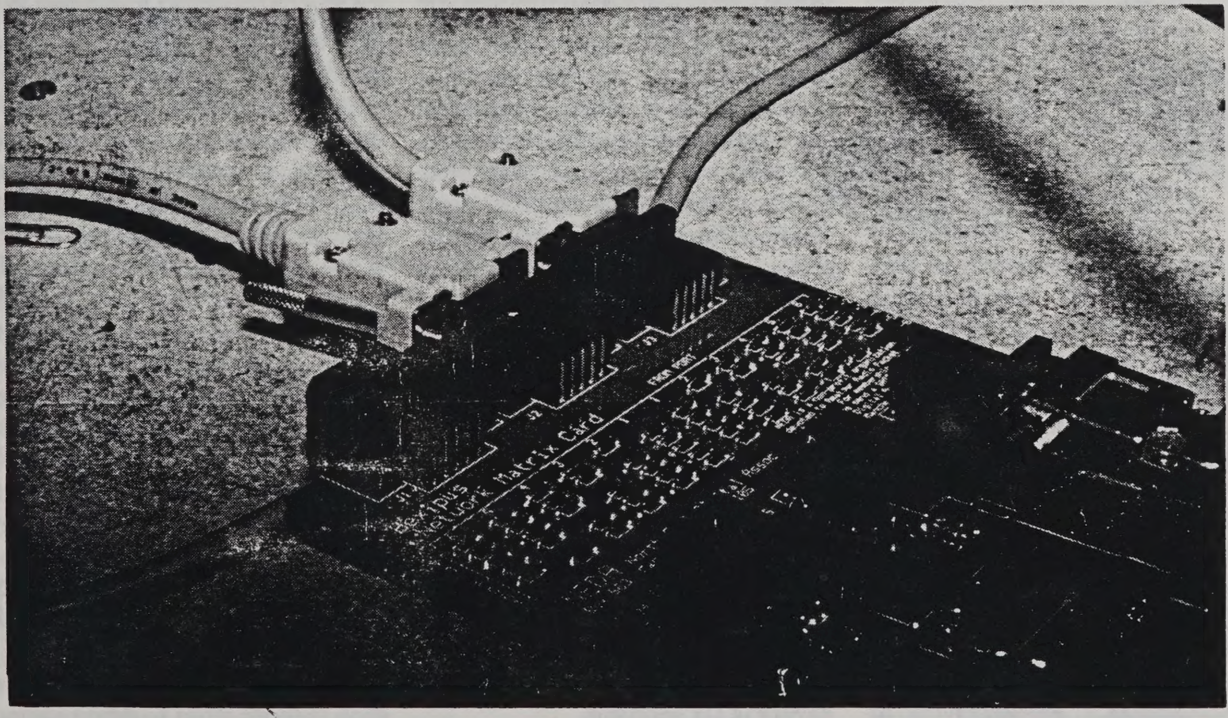
Network!

NEDA promotes operation on a multi-application packet network. The network is mostly TheNET based. Users may connect into the network via 2 meter access points using normal TNCs and access other users and servers over a range of four or five hundred miles. There are several dozen servers of many kinds available to any user anywhere in the system. There are many round table conference nodes called CROWD which the users meet on every evening. The most popular is CROWD at the CANDGA node site near Rochester NY.

Use of and growth of the network is encouraged. Users may connect to network nodes and observe how the system is configured. Membership is encouraged but is not necessary. Members have the advantage of being listed in and in receiving the Quarterly. Non members have access to this material only through privately photocopied copies. Generally members have more fun. This plus the fact of support of this process has been shown to be worth the membership dues.

The network is entirely privately owned or owned by radio clubs. NEDA does not own any hardware in the network. Network participants have agreed to certain principals that have been instrumental in seeing the system grow from a few sites to more than 70 sites. The following is a brief of those principals.

- The network is open for use by all packeteers;
- The network carries traffic for keyboard users, DxClusters, DxCluster users, packet mailboxes and mailbox users, playing of games, TCP/IP hosts, transferring of data or programs and many other kinds of operations;
- No user or server is more important than any other and must share the network equally (except for in emergency situations or in officially sponsored emergency drills);
- Network operators agree to a standard set of parameters, within the limitations of the software used, such that equal access to all is assured;
- Emergency operation and public service are of major importance, both because of the purpose of amateur radio and because by public service we get publicity and thus growth.



The NEDA HexiPus™ is only the first of the NEDA custom hardware projects. This board is used to connect up to six TNCs together to construct a TheNET node. NEDA payed to have 200 of these boards made. Rich, WB2JLR fronted the money and did the design work. The board is $29.95 + $4 shipping from the PO Box. There is an order form supplied with this Annual. Or send an SASE to the PO Box for an order form.

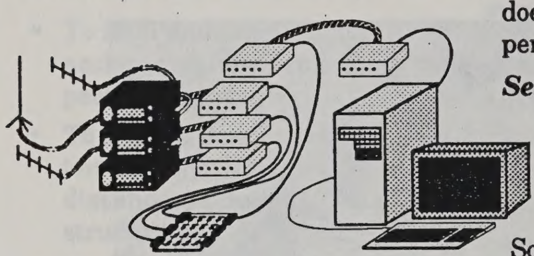
NEDA: What's a node?

A node is an active location in a network. A network is a collection of nodes which allow data to be carried from place to place. Each node consists of 1 or more ports.

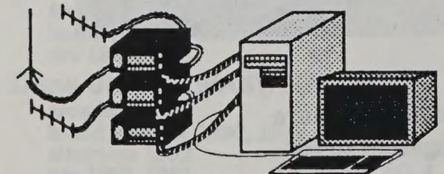
For this discussion I'll first break down types of ports and then try do give a brief description of the different kinds of nodes.

Ports

In most cases a port is where a radio hooks up to a node. If a node has two ports then it has two radios. Sometimes a node has a port that isn't hooked to a radio. The most common case is where the node is at the same site as another computer that is used on packet. The computer talks to the node by a wire link instead of by a radio link. In this case there is a port on the node which has no radio hooked up to it. The pictured node would be a four port node:



In many cases a node is constructed out of a PC. The radios may be connected to the PC via TNCs or may be connected directly to modem cards in the PC such as the case with DRSI cards. In these cases there is almost always an application running on the PC, such as a BBS or DxCluster. The application is not considered to be a port, even though that may be a destination of data. The pictured node would then be called a 3 port node:



How a port is seen by a user depends entirely on the type of software used for the node. See *Types of Nodes* for more on user interface.

What a radio talks to

Ports are described by what they talk to. What they talk to is described as users, servers and nodes.

Users

A user is a station which mostly accesses information from the network and sends short packets into the network. Personal home stations and EOC stations qualify. User stations predominantly connect into the network and access information. If a user is to send information into the network the information is sent slowly, as with a keyboard, or infrequently, as with a file or mail message transmission. Very few users send more than one file per day into the network. Most will send about one long packet every minute when they are very busy. Personal Message Systems (PMS) usually qualify as the PMS usually is sent to by a server from the network. The PMS doesn't send more than that one file per day, usually.

Servers

A server is a station that offers a service to more than one individual. Servers are connected to by users and by other servers. Some of the servers on packet radio today are *Bulletin Board Systems* and *Mail Boxes*. Both serve similar functions as message stores and file stores. Users connect to these servers and read bulletin messages, download information files, or send and receive messages with friends. Other kinds of servers are conference bridges which allow many users to communicate in a round table, and real time information resources which allow users to participate in the acquisition and dispersal of data. One common use for this kind of server is Dx spotting.

Nodes

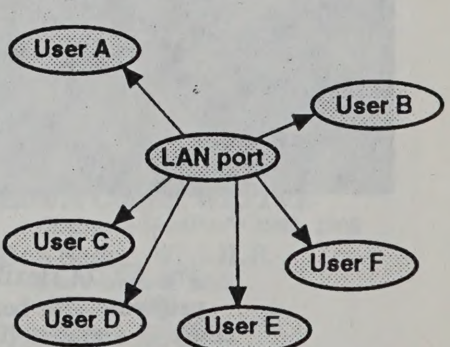
A node is a server that is used as a real time switch by many users and servers. As a real time switch any messages that are stored in a node are only there for a matter of seconds until they can be passed to a destination user or server, or to another node on the way to the destination.

Describing Ports

So, ports can be described as user ports, server ports or node ports, or a combination of the three. Good networking practice has it that ports should be configured based on the access requirements of the stations that it talks to, not on the type of stations they are. The ports are divided into two classifications: Ports where stations only receive data from the network (user LAN port), and ports where stations both send and receive data from the network (dedicated point to point link port). If best network design practices are followed, then for any kind of network node or server, these two port types are the only port configurations that need to be considered.

User LAN Port

A user LAN port is where a user connects to access the network. User LAN ports are used exclusively by stations that are either keyboard operated or that are receive-only stations, i.e. *users*. User LAN ports are on frequencies chosen to avoid co-channel occupancy with servers and other node ports. The user LAN port is very efficient in that there are very few incidences of collision. The reason that this is true is that if all users can hear all transmissions made by the LAN port, and users very rarely send data, except as an acknowledgment of data transmitted by the LAN port, then there will be very few transmissions that *could*



Even though *user A* can't hear *user E* they aren't likely to collide as they spend most of their time listening to *LAN port*.

collide. See *Hidden Transmitter Syndrome.*

Dedicated Point to Point Link Port

There are many ways to make a backbone. The most compromising way (and the most popular) is to put a radio on a frequency and label it a backbone. Other nodes and servers will have radios on the same frequency. In most cases there are radios using the frequency that can only hear some, and not all, of the other radios. This kind of backbone link will perform well only when the data throughput required is very low.

A better way of making a backbone is to make sure that all of the radios on the frequency can hear each other. In this case no radio will transmit when another radio is already transmitting. This provides a performance increase of better than a magnitude over the previous method. Most systems that are set up this way use a repeater to assure that all backbone radios can hear all of the other backbone radios on the channel. All but one of the sites would be a standard transceiver. The one site would be the repeater.

The best way of making a backbone with standard transceivers is to set up dedicated point to point links such that each backbone port talks to one other backbone port. If only two backbone ports are on a frequency in a given area the Tx/Rx cycles of the two nodes will toggle gracefully and throughput is maximized. The backbone port digital hardware is optimized for maximum transmit and receive response based on the other radios being used on the frequency. Experience by members of NEDA have shown that the performance increase seen using this approach is easily worth the increased investment of having a separate port and radio for each node to node link, over either of the two compromise methods.

Server Multi-access Port TCP/IP

In some networking situations it is uneconomical or unfair to designate stations which are part time as servers and force them to provide point to point links. This is the case where a station wants to operate as a TCP/IP host. A TCP/IP host is not likely to be happy on a LAN, forced to be a receive only station. TCP/IP is just too powerful and neat to be under that kind of restriction. On the other hand it is rather expensive to have to fund both ends of a dedicated link. Many would choose not to toy with TCP/IP *at all* if this was a requirement. Thus the Server Multi-access Port. This kind of port is operated in a hidden transmitter free fashion if it's going to work credibly. It is on 50MHz, 220MHz, or 420 and above. Because of the range limitations that are imposed by a simplex HTS free LAN builders tend to opt for a repeater environment as described above. There are now many TCP/IP environments successfully using a repeater. One of the TCP/IP stations on the LAN is designated to be the Network <-> TCP gateway if more than a few TCP/IP stations share the LAN. Only that gateway station is propagated using the TheNET protocol. That usually doesn't cause a major problem for the TCPers and it keeps the node routing tables short on the network. On a sparse network where there are few nodes in the node routing tables all TCPers on the repeater would use TheNET protocol to talk into the network.

Non Point to Point Backbones

In the case where a link is hidden transmitter free but not a dedicated point to point link, as in the case with a repeater, NEDA still calls the port a backbone port. The performance of the port will be different, especially when more that two sites on the frequency have data to send. To denote the difference between a dedicated point to point link and a non dedicated link channel the maps will show two different kinds of drawn line. Links that are not hidden transmitter free, such as between gateway ports or between gateway ports and servers, will be shown with yet a third kind of drawn line. See the example map elsewhere in this document.

User/Gateway Port

Because some areas have not yet set up dedicated point to point links between all of their nodes and because some areas are willing to suffer with inefficient network designs there will be areas of interface between the good network and the poor network.

Also, and more common, there is the case where LAN ports have gotten out of hand. All it would take is for a new node port to come on line within radio range of an existing LAN port, or for a server to operate a user access port on a LAN node frequency. In this case than LAN port becomes a disaster. Some classification for these ports is necessary. NEDA has decided to call this a *User / Gateway port.*

Types of nodes

Digipeater:

This is not referred to as a node by packeteers although technically it is a node. This kind of node accepts traffic from a packet station and relays the traffic to another packet station. Only traffic from one station to one other may be stored in the digipeater at a time. The digipeater will ignore other traffic until it can deliver the stored traffic. The digipeater does not acknowledge traffic handed to it, rather it is up to the destination station to take care of this. Thus if the digipeater is unable to deliver the traffic (the destination station gets QRM or signal too weak) it is up to the sending station to regenerate the data (retry).

Single port TheNET, NET/ROM, G8BPQ or MSYS:

This kind of node will accept a connect from a user station and allow the user to request a connect to another node or to another user. Data from a user is acknowledged by the node and then delivered to the next node or destination station. If the destination station or next node doesn't acknowledge the packet it is resent by the node.

Single port nodes broadcast lists of known nodes so that each node knows of all of the surrounding nodes. Taking advantage of this the user can connect to a local node and then request a connect to a 'next node' that may be several nodes away. The node will recognize the connect station as another node and will attempt the connection via whatever route is required to make the path. The path is usually selected based on the automated nodes broadcasts so sometimes the single port node may be tricked into using a path that is unreliable or might not work at all! (Covered in more detail later in this document) Most single port nodes are individual efforts and no system wide design philosophy is used so that many if not most paths between nodes are unreliable. It is always true, however, that a multi-node path will require traffic between the nodes. This traffic will be on the user's frequency, thus causing all of the users within range of the multiple nodes to be delayed. This also leads to tremendous occurrences of hidden transmitter syndrome (described later).

Single port KAnode:

This node is similar in operation to the single port TheNET and NET/ROM nodes except that automatic generation of node lists is limited to the adjacent nodes. This means that any connect to a 'next node' will not be via any intermediate node. KA-node does support automatic use of digipeaters between nodes however it is not compatible with TheNET, G8BPQ etc..

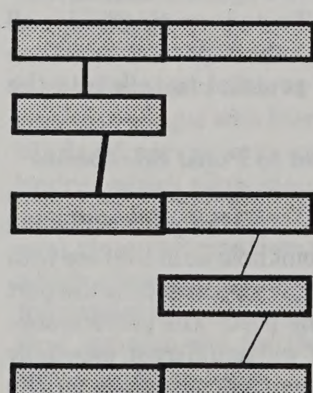
Multiple port TheNET, NET/ROM, G8BPQ, MSYS:

Like the TheNET and NET/ROM nodes described above this node allows the user to connect and then request a 'next node'. An important difference is that the 'next node' may be on a different frequency! These nodes consist of 2 or more independent ports, each port being a separate digital section and radio. The ports communicate via wires and are located at the same site. Connection between the ports is usually at 9600 or 19200 bauds (as compared to the usual speed of radio communications at 1200 bauds). Thus operation from one port to another at the same node site is nearly instantaneous.

Using multiple port nodes it has been possible to have packet users connect between frequencies transparently to access other users and automated packet stations (like mailboxes and bulletin boards). One idea that sprang out of this capability is the concept of the backbone.

Early Backbones

A backbone was taken to be a frequency on which two or more nodes communicated. This frequency would be other than two meters and would exclude keyboard users. This was so that there would be fewer 'hidden' sources of data. It was presumed that this would improve



the performance of the remaining stations and nodes.

NEDA has observed by experimentation that lack of performance on backbone circuits is proportional to the volume of data produced by hidden stations not the number of hidden stations.

Backbones: What happened?

What has happened is that each region would set up a backbone channel such that long haul traffic could be moved off of 2 meters. Some of these backbone channels were set up by bulletin boards (before TheNET and NET/ROM) so that traffic could be sent to other BBSs without interference from individual non-bulletin board stations. A popular example of such a backbone implementation was the common use of 221.11 and 441.0 in the north-eastern states.

Since the advent of multiport nodes it has been possible for traffic to pass from one regional backbone to another in a single point to point connect. I.E. a station from Connecticut could connect to a station in Maine by connecting from his 2 meter radio to a node that had 220 capability which would talk across 220 to a node that had both 220 and 440, and then across 440 to a node that had both 440 and 2 meters, and then back to 2 meters. As time went on each of the backbone frequencies (and there were only several) has gained in quantities of nodes and quantities of data. Originally there was no attempt to control hidden transmitters and precious little to control erroneous path generation due to unbalanced transmitters verses receivers at each site. (Automated node broadcasts, remember?).

NEDA network node

A NEDA node consists of all of the interconnected TNCs, computers, radios and associated hardware at the single site, which performs the switching functions for it's piece the network. The definition holds true for whatever type of networking software in use. I.e. NOS, MSYS, TheNET, NET/ROM, ROSE, TEXNET, G8BPQ etc..

These nodes have multiple ports, at least one of which is a backbone port. The backbone ports talk to

other backbone ports such that packet data can travel from node to node on non user frequencies. (This way users are the only stations that have to share user frequencies) The important difference between a NEDA network node and most other multiport nodes is that the backbones in the NEDA system are maintained as hidden transmitter free point to point links (See page on hidden transmitters later in this package). This is done simply by supplying a separate set of radios on an independent frequency for each backbone. In concept this is extremely simple and obvious. This incurs several disadvantages however.

NEDA node: Disadvantages

The first is obviously cost. Each of the sites houses at least 2 sets of radio and TNC, most sites must have 3 or more. In today's network two of the sites have 8 sets of TNCs and radios. Most have 4 or more sets. That's a lot of radios and TNCs.

The second disadvantage is that each frequency must be chosen to not have interference from any other station, at a different site or at the same site. It's difficult to have 2 backbone frequencies coming into a single site on frequencies that are near each other. There are really only 2 bands on which backbone links are conveniently constructed, 220 and 440. If a site has to have 3 or more backbones it is important to maintain frequencies that are separated when in the same band and radios and antennas that will not interfere. This becomes a technical challenge. Also there should be a different frequency for each backbone in a given region. That becomes an administrative challenge. In order for the system to work well there cannot be other stations operating on the backbone frequencies. That becomes a public relations challenge.

NEDA node: Advantages

So, why bother? The first answer is performance! There is at least a 5 times performance increase using the same baud rates and hardware as the old style 'backbone'. If only two stations are talking to each other across the backbone the timing values in a TheNET or equivalent node may be maximized for that situation. That can lead to a 20 times performance increase, or more, over a non hidden transmitter free backbone. That's a 20 times improvement (at least!) with a 1.5 times increase in site cost. Not too bad a trade off.

The second answer is marketability. Once built in the way described in this document a network is usable by keyboarders, Dx spotting networks, BBS stations for forwarding, users for access to BBSs, TCP/IP hosts, etc. Because network site to network site transport is moved off of 2 meters that band may be used for low power personal keyboard stations. The amount of fun and interested had by all of the network users and participants goes way up. A 60% of all hams involvement rate is achievable. This will allow a very successful packet program to exist.

The third answer is adaptability. It is easy to replace or reconfigure all of the ports on a single backbone channel with many fewer complications because there are only a few sets of equipment involved. A performance increase on all ports involved could be achieved by adding 4800 baud modems at each of the few sites.

It's not really that hard to add additional ports to a single or dual port node. For example at sites with limited antenna space it may be possible to use dual band and diplexed antenna systems to achieve multiband operation. Most backbone links run relatively low power and directional antennas which make it easy to keep the transmitters out of the receivers.

Backbones using repeaters

Instead of supplying separate sets of hardware for several backbone channels at a site it is possible to do backbone linking through a repeater. The repeater would be placed at one site and then several others would access it. There are several advantages and disadvantages to this.

Repeaters: Disadvantages

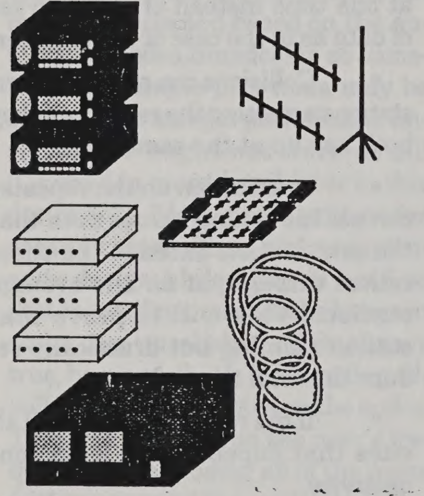
- Each site that links to the repeater must have a dedicated radio on the same frequency pair as the repeater. This ties up two frequencies and the same pair must be used at each site. This is instead of being able to choose a different link frequency from the 'main' site (that would have the repeater) to the other sites based on the spectrum availability at each of the sites.
- The total data throughput (added for all site'rptr links) is going to be less than the baud rate (which must be the same on all links to that repeater). If the 4 links were on independent frequencies the throughput is theoretically N *times* the baud rate for a system of N sites. This is because a different set of data can be traveling on each of the links at one time instead of only one set of data as in the case of the repeater.
- Collisions can occur between stations accessing the repeater if they both key up at the same time.
- One station on the repeater can set his timing values such that the one station excedes the theoretical throughput for the average station. This will improve that station's timing but drastically reduce the total throughput.
- If the repeater should fail all sites that depend on it loose connectivity.

Repeaters: Advantages

- The total packet travel time on the repeater, if not overloaded, would be half that of a separate link system.
- The cost of the system is one full duplex radio and 1 normal radio per site as opposed to 2 radios times number of sites.
- If a central link site is neccesary and that site is a commercial radio installation it might not be practical to have several link radios. A repeater only needs one antenna.

Note on repeater usage: All stations involved in the repeater operation must be using modem tone DCD, not noise level DCD unless the repeater has a very short keyup and unkey delay.

NEDA recommends that repeaters be used on user ports *or* for tying part time or low usage servers into the network where the servers aren't feeding other servers. In other words, if a server has only one port then it could be tying into the network via a repeater. If a server has two ports then it should be using a point to point link to get into the network. For user access to the network repeaters are excellent but only if the repeater is exclusively for low duty cycle users. Users sharing a repeater with a server is not a good idea.



What does it take to make a N.E.D.A. node?

The most common node configuration consists of:

- two UHF or 220 FM radios of the 25 watt power range,
- a 2 meter radio of the 10 to 25 watt power range,
- two yagis,
- an omni,
- three TNCs,
- three runs of coax,
- a power supply,
- a HexiPus™,
- miscellaneous serial cables, mike cables and power cables.

The omni is 6dB gain or less for the 2 meter user port.

User Port Recommendations

Most sites have a user access simplex 2 meter user port. The user port is geared to cover about 50 packeteers. If the normal ratio of hams to non hams is 1:600 and the normal ratio of packeteers to hams is 1:2 then the population area that your user port should cover should be less than 60,000. If there are 5 frequencies available on 2 meters then a city of 300,000 could be covered with very little concern for overcoverage. Be sure to take rural areas into account when calculating coverage. It is important that the number of packeteers (stations that receive-mostly) that must use a single user port is 50 or less. If more than 50 need to access the user port than attrition of packeteers will result. Thus it is important to watch for over coverage. It is also important that some frequencies are left available for expansion and experimental operation. It would be real silly of us to assume that we're using the *best* possible networking tools. Some frequencies must be left on 2 meters for high tech LANs. It should be stressed that 'high tech LAN' does not include a wide area coverage *mess*.

If a situation exists where there are more than ƒ× 60,000 population a directional antenna might be used, or attenuators for reduced range. Also keep in mind that the more LANs in the area, the better. If someone wants to put up a node where there is already LAN coverage then space *must* be made, by dividing coverage with the aforementioned directional antennas. Telling someone to not put up hardware when they offer is as good as telling the FCC that we don't need an amateur service after all. The thing to do is to tell the newcomer *how* to use the hardware, not *whether* to. Think up some good way to use the equipment without compromising the network and without hurting the other users. Show them this book. Get it together!

Backbone Recommendations

Each site has at least one hidden transmitter free backbone port that talks to another NEDA node. The backbone port is on 50MHz, or 220MHz and above. Most of the backbone ports use yagi antennas pointed at the neighbor node to which the backbone runs. This way we have some hope of reusing backbone frequencies within the network. It is also desirable for practical and financial reasons to put more into the antenna so that sensitivity and output power may be minimized. The backbone port must be running software that is protected against unauthorized operation. The software we've been using mostly in the network is TheNET Plus v2.08B by Bill Beech, NORD><LINK and others. The parameters used in a backbone node are optimized for each end of the link. This allows us to often have less than 300ms transmit/receive overhead on a backbone link.

Aux Port Recommendations

Many of the sites have an additional auxiliary port on 220 or 440. This way stations who want to experiment with running a server or stations who run part time servers or TCP/IP hosts may have a non 2 meter frequency to access the network. As it is considered grossly unfriendly to send lots of data into the network via 2 meter ports the only way a station could operate a data generating station would be via one of these open 220/440 user ports or via a point to point link.

The most successful NEDA nodes have all ports based on TheNET software running in a PAC COM Tiny 2, MFJ-1270B or other TAPR2 compatible TNC. It is quite possible that in the future nodes will be based wholly on other kinds of equipment.

Node Construction Resources

NEDA has developed resources to help in the construction and debugging of nodes. Contact NEDA for more information on the latest. One such product was the Tjp Octopus which was used to tie the backbone TNCs and user port TNCs together via their serial ports. The Octopus card is a diode matrix card that is

necessary to hook up PAC COM and MFJ TNCs together in a node cluster. The Octopus card was created by N0NDO, John Painter at recommendation by KA2DEW, Tadd and WA2WNI, Dana who then became NEDA founders. John funded the Octopus card and sold the card personally. PacComm at one time took 50 of the cards on consignment. The Octopus was replaced in January 1991 by the HexiPus™ Card which was designed by WB2JLR, Rich Place. NEDA now sells the HexiPus™ for $29.95. See Order Form in this document.

Recent vintage TNCs may be tied together at 9600 or 19.2K baud (only the PAC-COM will do 19.2K baud) across the matrix. The matrix is unnecessary for a 2 port node.

So, to make a 3 port node (user port and 2 backbone ports) you'll need the following:

2m rig, gain omni antenna, coax.

440 backbone rig, end mount beam or dipole, coax

220 backbone rig, end mount beam or dipole, coax

3 TNCs, 3 node chips,

HexiPus™, cable hardware (connectors included with PAC-COM TNCs)

Power supply and control point mechanism.

Other Comments

Node sites don't necessarily need to be located on a big hill. It is important that it can serve a specific geographic region of *users* with its coverage effectively and *not* interfere with adjacent packet systems using the same frequency. Some systems that are at high elevations tend to "hog" a given user channel over a larger territory than necessary. A better approach is to create small local area networks (LAN) with well defined areas of coverage. This also allows for the reuse of 2 meter user channels at nodes that are closer together without interference. Further efficiency is achieved by the fact that fewer users will access each user port. This is called the cellular approach to user ports.

Node site accessibility is also an important consideration for constructing a node. There are some serious advantages to putting nodes at the node owner's house:

- The node can be serviced in short notice and with little hassle.
- The operator can observe problems that might not be apparent from a remote station.
- Radio equipment that is not hardened for an outdoor environment will work fine at a home node location.
- The site manager may attach other systems to the node via a wire link (as opposed to radio link) such as a BBS, TCP/IP station etc.
- Diagnostics may be done to the station using equipment that one might not want to haul or leave at a mountain site.
- Node reconfiguration for experimental or emergency linking is possible.
- Christmas lights are unnecessary as the TNC light show is fantastic
- The node is available for demo for curious visitors.

Who Builds NEDA Nodes?

In the "early days" of the NEDA Network, implementation of new nodes was strictly a word of mouth affair. A node op would bump into another interested packeteer on his keyboard while hacking about the known universe and in the course of discussion the new packeteer's interest would perk up. Off the shelf hardware and networking technology was enough and before you knew it another node was on the air!

As the network grew and the excitement of it all started to spread NEDA slowly became the buzz word of much of the north eastern region. Most BBS sysops quickly found out because some serious improvements to traffic forwarding and bulletin distribution took place in leaps as new NEDA backbone links came on line. Soon there was connectivity and the capacity for good data throughput from Kingston NH to the Central NY region. Expansion was on a roll. In fact things were expanding so fast that it became difficult for NEDA contact people to keep up with all the requests for information on how to link to the NEDA Network and acquire the technology ideas that were being passed around!

The next thing that occurred was not really expected, but in hindsight it is easy to understand why it happened. From the outside NEDA appeared to be very large, covering several states and encompassing about 3 dozen major node sites. It sure looked like NEDA was busily sticking up *nodes* everywhere! Thus imagine the surprise of NEDA members and node ops when they went to a major hamfest and discovered that the rumor mill had created an atmosphere of "NEDA's coming to town! They're gonna put up our *node*!".

This in essence became the "*NEDA Myth*", that is, when NEDA gets around to addressing networking development in **your** area that NEDA would purchase, install and operate the local node!

This is very much not the case! NEDA does *not* "come into town and put up *your node*". What really happens is that hams who want to make it happen ask NEDA members to share information resources with them and then put up their own node. Of course all the other NEDA node ops are just as anxious to have new participants in the network and will be right there to give you ideas, technical information and support as you put up the system that will serve you best and link it into the rest of the network.

So what is NEDA? NEDA is the group of individuals who have "*made it happen*". NEDA would like to encourage everyone who wants to become involved to not wait for

NEDA to "come to town"; if you are reading this now and you are a member of NEDA, then *NEDA is already in your town*!

If you are interested in hearing more about node construction or if you are planning on putting up a node, whether in the NEDA network or not, you should contact the NEDA technical committee. Send a packet to NEDA @ K1MEA with ATTN: Tech Committee in the title.

Network Conception

Important Considerations when first specifying network hardware and direction include:

- Network capacity;
  Can the network handle the capacity that will be imposed on it? This is partly determined by the existing network as the users will generally only expect what they have already gotten used to.
- Network promotability;
  A good network is of little use if the target audience can't understand it.
- Operator and technician availability to fit the chosen hardware and software;
  Using equipment or software that is obtained or built only with special talents or connections to build the prototype network will not lend to a successful network. So, either choose off-the-shelf components or start your network development by creating sources for the required materials.
- Politics;
  Unfortunately this is an important consideration. In order for your budding network to survive it must allow for the participation by people who are greedy and egotistical. The network design must allow avenues for those people to help the network, without destroying it. In the same token it's important that no design rules are made which give special privilege, especially where this may be construed to be negative by any party.

NEDA suggests that in any new network expansion or project in a new area that the links be put in at 1200 baud using point to point links for the following reasons:

- 1200 baud is misconstrued as being slow, especially by those who are only used to non-point to point links. A 1200 baud point to point backbone with 250ms keyup delays can pass 8 Megabytes per day.
- Any network linking equipment that is installed and working is worth substantially more than just the parts. That means that you can turn around and reuse the equipment somewhere else if you upgrade an existing link. Waiting to get faster equipment before putting in a link is silly. Put it in slow, first, rather than wait. Then upgrade as resources become available. A 1200 baud point to point link is far better than a HTS infested link channel.
- If you have a network that is promoted to all hams, you will find that some of them will desire to get involved and expand your network. This will lead to more network facilities and redundancy. The more people building on the network the better it will be for all.

Basic Networking Guidelines For An Amateur Radio Packet Network

If these rules are followed the participants will create a fun, expandable and upgradable packet network that will be the equal of any in the world. Compromising on these in any way will lead to limitations and eventual dead ends. These rules apply to amateur radio in the United States and may see some modification in other countries merely due to spectrum changes and government regulations. If a system is going to be created and used by hams, and depend entirely on ham radio, then these are good rules:

1. All backbones are dedicated point to point links. Backbones are on 50MHz, 220MHz and up. Backbones are *never* on 144MHz. See *Hidden Transmitter Syndrome.*
2. User LAN channels have no servers. Just one node for access to the network and no more than 10 active users at a time during a standard operating period. Users connect to the node and then away from the node, either via the network or direct from the node to another users. No digipeating.
3. Servers connect to the network via dedicated links if they are high volume data generators or via hidden transmitter free shared links if they are not high volume data generators. Server links are on 50MHz, 220MHz and up. Server links are *never* on 144MHz
4. User access to servers is via the network. Servers shouldn't have any 2m hardware as they are connected to via the network. An exception is where users can gain network access via the server in which case: the server would be also be considered a user access port; the server frequency is not on the same LAN as any other user access port; users can get to the rest of the network from the server's user port. In any case servers don't need 2m hardware on the same frequency as another user access port in the same locale. (Except for redundancy)
5. Corollary of 2. LANs are designed so that they do not see other nodes or users of other LANs. LANs need to be low power so that this can work. No node to node communications may exist on 2 meters. See *Hidden Transmitter Syndrome.*
6. Locations and mechanical design of node housings are not important. Nodes may be on mountains or in homes. Backbone and dedicated links may be of high or low power depending on need.
7. Length of backbone hops is not important. Reliability and signal to noise ratio are important.

8. All nodes/users in the network must have standardized window sizes. They don't have to be the same but they must be agreed upon by all network level 4 data sources. This is important because otherwise network users will not have the same priority.

9. Any station that will transmit data at greater than 300 chars/minute is a server. Any station that provides services to stations over the radio is a server.

10. Link and backbone throughput should be improved as loading increases. 1200 baud is pretty good for backbones in a new network, if links are point to point, dedicated, and with no interference. See *Node Radio Considerations*.

11. Radio interference from on site equipment is just as bad as interference from off site equipment. It must not be tolerated.

12. Redundancy is important.

These are good rules for any amateur radio packet network using AX.25 based link layer protocols. Additional restrictions may be imposed by network software. Any comments on these rules should definitely be aired. Times and technology change. This is a good start though.

TheNET Network Development Guidelines

1. Time-to-live should be established network wide. Link qualities should be set so that the furthest propagating node is not as far as time-to-live. See *Node Propagation*.

2. Windowsize parameters should be the same and should be a low number. The value should be such that during 50% loading the number of packets outstanding for a given connect is about 1 per 10 seconds of planned latency. This allows for approximately 23 characters per second through put for each connect. With 100cps throughput rate on backbones and a time-to-live such that three 1200 baud backbones are the maximum traversed in a L4 hop a windowsize of 2 is appropriate.

2. Nodes broadcasts should be turned off on LAN ports to discourage node to node communications on 2 meters.

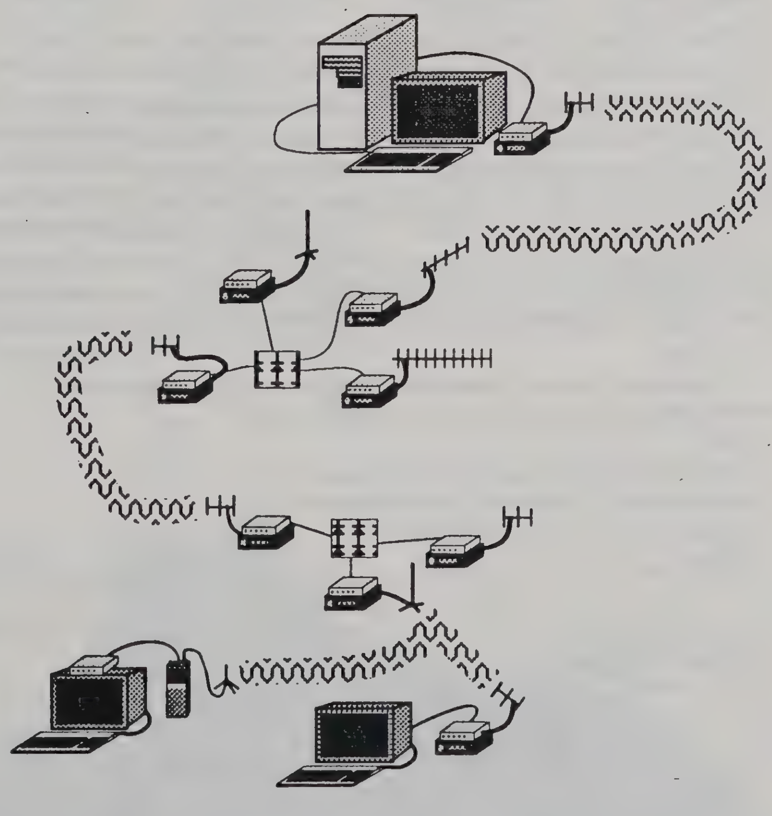
3. # node propagation needs to be turned off. This is because of the limits on node list length.

4. Retry rate/level 4 time-out should be the same for all nodes and should be greater than the worse time it takes to transmit the maximum length message the maximum number of nodes allowed by time-to-live and over the worst path in the network.

5. Nodes whose L4 retry rate, L4 windowsize and time-to-live are not predictable must not be allowed to take part in network level 4 operation. All L4 network hardware operators must agree on these figures. This makes sure that all network users get as close to an equal share of the network capacity as possible.

Level 4 operation means that the node can pass data to another node, multiple hops away. If a station is denied level 4 access then it must do a simple connect to the adjacent node and then connect on into the network. This might be necessary in the case of a non-compliant TheNET, MSYS, NOS, G8BPQ.

6. Nodes broadcast limitation and nodes broadcast reception control are used to limit the level 4 access of nodes that are not network participants (i.e. who don't abide by the above rules, including rule 6).



How To Use The Network

Let us take as an example a new user. The user wants to get to a friend in Poughkeepsie New York whose call is K2QRM. Our user is located in Exeter NH. Assume for the sake of the example, that both are using 2 meter stations and have base station antennas. The station in Exeter is looking at his NEDA user port map and finds that KNGSTN:WB1DSW-5 is on 145.05. He dials his radio over to that frequency. Next he tells his TNC to connect to WB1DSW-5. He gets a message on his display that says:

```
*** Connected to WB1DSW-5
```

At this time, out of curiosity he asks the node for its INFO message and a list of nodes by doing the following:

```
I
KNGSTN:WB1DSW-5}
sysop WB1DSW & NR1N
  QTH East Kingston NH
 freq 145.05
 info NR1N@WB1DSW

download file NETWORK in the N directory
( DN NETWORK ) on WB1DSW BBS for more information about local nodes
```

then he types:

```
N
KNGSTN:WB1DSW-5} Nodes
```

ARLNG1:K1CF-1	ARLNG2:K1CF-2	ARLNG4:K1CF-4	ASH:KB4N-5
BBSASH:KB4N	BBSDSW:WB1DSW-5	BBSEDY:KA1EDY-1	BBSGOZ:KA1GOZ-1
BBSN:NS1N	BBSPHY:WA1PHY-1	BBSWOK:WA1WOK-2	BED:WA1PHY-3
BELNAP:N1DCT-3	BERK:WA1TPP-3	BERK2:WA1TPP-13	BERK3:WA1TPP-14
BERK4:WA1TPP-9	CENTNH:K1BKE	CHSTR:K1MEA-2	CROWD:WA1WOK-7
DENNIS:KQ1K-7	NASHUA:KA1GOZ-9	NBY:KA1EDY-5	NHOEM1:WA1WOK-1
NHOEM3:WA1WOK-3,	NSHORE:KC1PK-5	NSHR22:KC1PK-3	SALT:KQ1K
SCIT:NS1N-5	SWNH:KA1BBG-1	STMFRD:KB2CS-1	SWNHU:KA1BBG-4
WNDHM1:K1TR-1	WNDHM2:K1TR-2	VNH:WA1TLN-1	YCCCDX:K1TR-3

The Info message can be up to 160 characters long and is set by the node manager. All of the N.E.D.A. nodes have useful info messages. The nodes list is a table of node names and callsigns of the nodes that are available via the backbone within 3 hops and in some cases the nodes that are heard by the user port on the user port frequency. Referring to the map our user sees that CLV:N2CJ-1 is a node near Poughkeepsie NY. The route from KNGSTN to CLV is KNGSTN -> WNDHM -> CHSTR -> STMFRD -> KNDRHK -> CLV. Each step is via HTS free backbone links. Because the system only shows nodes three hops away the user must step to a mid-point node. STMFRD shows in KNGSTN's nodes list. He then types:

```
C STMFRD
```

if all goes well about 15 seconds later he gets the message

```
KNGSTN:WB1DSW-5} Connected to STMFRD:KB2CS-1
C CLV
```

and after 30 seconds gets the message

```
STMFRD:KB2CS-1} Connected to CLV:N2CJ-1
```

At this point our user can type

```
C K2QRM
```

and after about 35 seconds gets

```
CLV:N2CJ-1} Connected to K2QRM
```

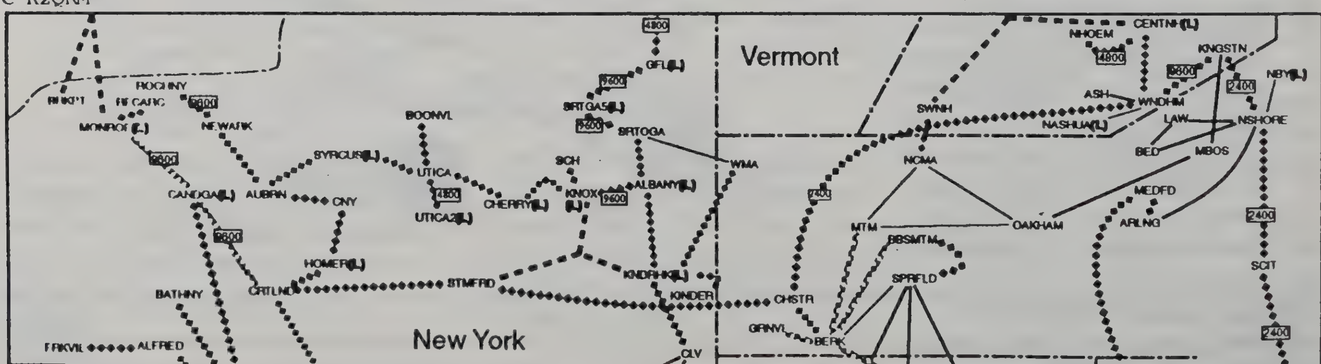
Now any text typed by our user will go to K2QRM and any text from K2QRM will come back to our user. To terminate, either party simply exits back to command mode using control C and types D as usual.

If our user desires he may connect back to KNGSTN node and then to STMFRD and type an N command. This will show the nodes that STMFRD knows about from it's point in the network. Note that the user ports on KNGSTN and CLV (as well as STMFRD) may be on different frequencies. It is only important that the frequency of the user port is the same as the users that is talking to it. The network is responsible for connecting user port to user port. In this case KNGSTN is on 145.05 and CLV is on 145.09. Even if they were both on 145.05 the network would still be used to pass the traffic. Also in this case KNGSTN and CLV are about 200 miles apart. At each hop the data travels from node to node via the backbones. Each backbone is on a separate frequency. All of this is transparent to the user.

When you establish a frequency for your station you should make note what network user port serves your station best. You can then tell your friends how to get to you from the network.

This document's purpose is to provide the information necessary to build a network of this type. Also included is more information on using the network, both as a user and as a server.

Happy Packeting!



Hidden Transmitter Syndrome

This is the bane of most earlier packet networks. A system with 3 sites: Station A and Station C are far enough apart that they don't hear each other at all. Station A has traffic to go to station C and station C has traffic to go to A or B. Station A will transmit when it doesn't hear anything. Station C will do the same. Site B hears both A and C. If C is transmitting and A decides to transmit, both messages are lost. If A is waiting for a reply from B and C is talking, then A has to wait. If C is talking for too long, A will retry, thus trashing the message C is sending to B.

If the A to B link was on a different frequency than the B to C link, the observed performance increase is greater than 5 times, regardless of the baud rate! *A hidden transmitter is a station that can be heard by one or more stations on a frequency but can not hear ALL of the stations on the frequency.* It is the stern recommendation of N.E.D.A. to not allow hidden transmitters on backbones.

Hidden Transmitters on User Ports

Clearly it is not possible to eliminate HTS on simplex user ports. It is certainly possible to eliminate HTS on a user port if the user port could take advantage of a duplex repeater. However, given that most user ports are simplex systems the effects of the hidden transmitter problem must be taken into account

If the user port node is the only station on frequency that can hear everybody and if all stations on frequency have parameters set to take advantage of this, the following is true:

Given that a user port can be heard by all stations on a LAN: *Time used for data transmission by the node is about 80% efficient. Time used for data transmission by user stations is about 80% efficient, under minimum load. Under maximum load the time used for data transmission by the node is still 80% efficient but time used by the user stations drops down to near 0% efficient. Thus it is useful to have sources of lots of data (i.e. automated stations like BBSs) to be sourced through the network and not from a normal user site on a 2 meter user frequency.*

For this reason NEDA requests that stations which source a lot of data (i.e. servers, hosts and BBSs) use dedicated link channels on 50, 220, 440 or above. This allows for the best of all worlds for the servers and for the users. If we create our network in this fashion we'll create a system which will grow and in which fun and learning are maximized.

Minimizing HTS

1. On Backbones:

Keep each link hidden transmitter free. Try to configure backbones as dedicated end to end links using only two radios per frequency/link if on a high traffic path. High volume means that there is channel activity as much as 1/4 of the time during peak loading periods.

Set persistence values at each port on a backbone to 256/(N-1) where N is the number of nodes on your backbone frequency. Example: If 4 nodes on backbone frequency then use Persistence of 85 at each port.

Set slot time at each port on a given backbone equal to TXDelay for that port.

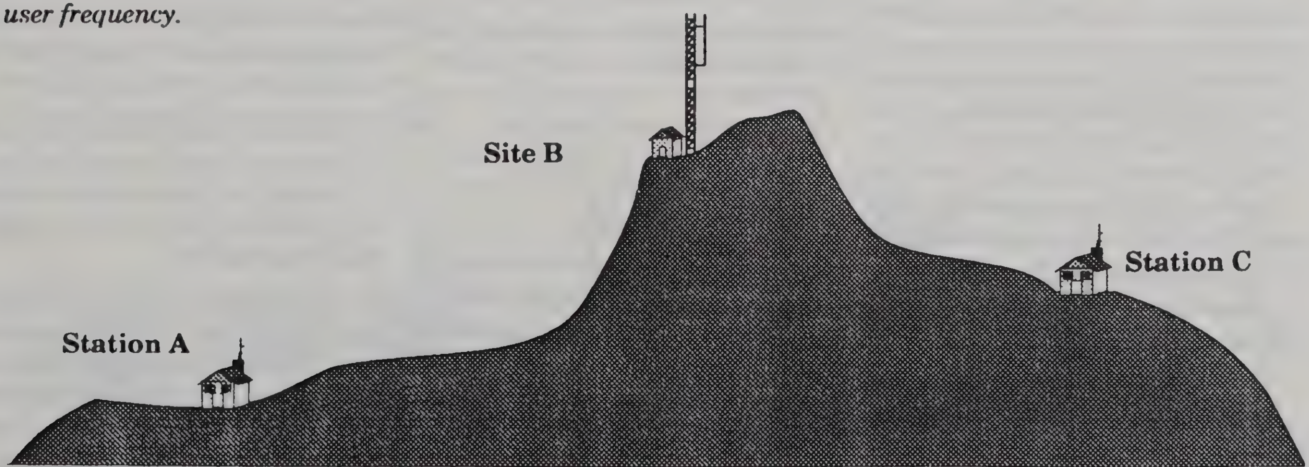
2. Users that are high data volume sources:

High data volume sources should minimize collisions with low volume users by setting up dedicated uplink channels to one or more local nodes. Arrangements might be made with other local high data volume sources to share a dedicated link port. This should not be on 2 meters.

3. Users that are not high data volume sources:

These users should arrange to use minimum power to assure full quieting signal at a NEDA node user port which has few or no co-channel nodes or servers. Observe the node in monitor mode and see what/who is uplinking to it. What/who is downloading is much less important. On channel DXing of user ports is seriously not a good thing.

Keep MAXframe low (1 or 2), and DWait at 16. CROWD conference nodes let you use PACLEN up to 240 so set it at 240.



TCP/IP

TCP/IP is one of the protocols that run across the TheNET network. The way this works is that TCP/IP hosts have dedicated links into TheNET node sites on 220, 440 or up. Then the TCP/IP host transmits a TheNET node ident into the network. The Idents all start with the two characters 'IP'. Thus IPWMA, IPQRP etc.. That node ident propagates up to 7 TheNETs away to all other similarly linked TCP/IP hosts. TCPers that want to access hosts across the network must route through TCP/IP hosts along the way. No end to end TCPing is supported as the node broadcast qualities in the TheNET network are only 7 hops. For new TCP systems where this would be a problem, compromises, either involving software hacks or additional appropriately located TCP hosts may be constructed. Contact your local NRC or WA2WNI for more on this. Also check the user port maps for the locations of TCP hosts in the network.

Note: Any TCP host link into the network must coordinate TheNET-NET/ROM parameters with WZ2B. Thanks.

Some TCP stations access the network via TCP-only links into hosts which are already linked into the TheNET network. Additionally it is possible for non-TCP stations to access TCP hardware over the TCP-only network by doing TheNET connects into the TCP hosts that *are* listed in the nodes lists.

The recommended network subnet for TCP/IP usage is that one or more TCP/IP hosts would have point to point links into TheNET nodes that are in the network. Those TCP/IP hosts would support a TCP/IP-only repeater. What this does is set up a subnet for TCPers which would be entirely TCP/IP. That allows the TCP stations to use backoff and P-persist without competing with AX.25 non-TCP stations. Additionally it allows the community of TCP stations to interact so as to further build packet activity. We should stress that backbones should be a cooperative venture between all packet users. It is probably true that there are other schemes which would work better on the long haul than TheNET. However, TheNET is the only protocol currently available that supports all kinds of packet users. If there is new information on this please present it to the NEDA Technical Committee. Why don't you join that committee while you are at it?

Note to computer hackers: UHF link radios need not be expensive. All you need to do is find a RF hacker to help if you don't want to mess with it. There have been two separate UHF radio notices in the Quarterly for under $80 per rig. NX2P also sells a UHF link radio (from PacComm) that would do very well. It's in the $200 price class per rig.

What is TCP/IP?

Transmission Control Protocol/ Internet Protocol. TCP/IP defines a protocol suite. TCP/IP is a system of messages sent between computers, via radio (or telephone, or wire) that enable the computers to exchange data meaningfully. Where AX.25 is a protocol that defines how two TNCs can communicate, either directly or via digipeaters, TCP/IP is a protocol suite that defines how two computers can communicate, over wire line, telephone with modems, two or more TNCs, NET/ROM, TheNET, ROSE, etc. TCP/IP is called a suite of protocols because it actually includes hundreds of different message types and response procedures for dozens of different purposes.

Defined in TCP/IP as commands (and separate protocols) there are TELNET, FTP, SMTP, FINGER, PING and others that are of direct use to the user.

TELNET establishes a real time two way interactive connection between a user at his own computer and another remote computer. This lets the user command the remote machine as if he were sitting at the keyboard of the remote machine. This is similar in effect to how an AX.25 user perceives TheNET and BBS operation.

FTP or File Transfer Protocol is a customized command set for getting or putting files on a remote computer from the user's computer. Files may contain non-text information. This is a key feature of TCP/IP for amateur packet radio.

SMTP or Simple Mail Transfer Protocol is a system for automatically routing multi-line messages from one computer to another over any number of intermediate computers. Unlike FTP and TELNET which require that an end to end path must be established in real time SMTP allows messages to traverse the computers that are available and then wait for computers that are unavailable and then proceed when they come on line.

FINGER is a command whereby a user can ask a remote machine for information about another user. Thus I could do a finger on the user NQ1C on the machine NQ1C and get back that NQ1C is Bob Lafleur and whatever other information that Bob wants his finger file to contain.

PING is command to send a packet to a remote computer to find out if it is connected to the network and if so how long it takes to get a packet there and back.

There are many other useful protocols built into TCP/IP that allow such things as data sharing between programs running on two different computers, identifying what hosts are available, finding out the time at

a remote machine, authenticating passwords and even passing silly quotes.

Message routing with TCP/IP is based on a 32 bit address and aliasing. Each host computer is given it's own specific 32 bit network address and a text alias. The text alias for amateur TCP/IP is usually callsign.ampr.org. The .ampr.org is used to differentiate the amateur network with the commercial networks in cases where there are tie ins between the two. The 32 bit address if of the form 255.255.255.255 where each of the numbers is called an octet signifying that it uses 8 bits. Each of the four octets from left to right decreases in priority. The first octet is used to determine whether the destination address is ham, military, commercial, educational, etc.. The second octet might indicated which state the destination machine is in. The third, depending on how the ham TCP/IP addressing committee decides to run things, might determine a network node output or a county or city. The last octet determines which individual machine that the message goes to. So, given that the network extends across the country, it should be possible to address a message from any TCP computer to any other. The addressing system also allows for more than one user at each machine. Thus I can be ka2dew@k1tr.ampr.org which is different than k1tr@k1tr.ampr.org.

The process in the TCP program which sends messages from the host and waits for acknowledgment from the destination station is more sophisticated than TheNET. With TheNET up to four messages are sent out of the originating node and then when acknowledgments come back for those messages new messages can be sent out again. Four messages may be outstanding at a given time. If an acknowledgment for a message doesn't come back for 5 minutes the originating node will regenerate the message. With TCP/IP what is a 5 minute timer in TheNET is automatically adjusted depending on previous performance of the link. This is called 'backoff'. TCP/IP is loaded with this kind of intelligent networking features.

TCP/IP using the KA9Q software package is very easy to modify. NET.EXE which is the original KA9Q package has been added to by dozens of other amateurs. NOS, Network Operating System, is updated and customized by many hams for many purposes and is entirely public domain. This is in contrast to TheNET which is only modifiable with great difficulty.

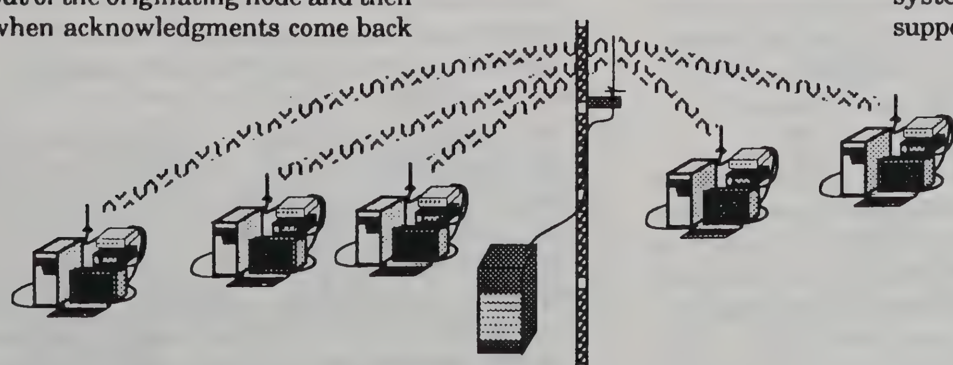
TCP/IP is a mature protocol system due to the vast number of people working with it. TCP software is available for most computers. It can be run over a huge number of different kinds of data links. It is extremely powerful. It is in use on many, if not most, commercial workstation systems in the world. Sun Microsystems, Apollo, HP, Xerox, DEC, Apple, Next, Wang, and most other computer companies either use TCP/IP exclusively or at least offer support for it as a standard feature of their computers.

Why Aren't All Packeteers Using It?

The first reason is that it won't run in a simple TNC. It is quite possible to have lots of fun and take advantage of many of packet radio's capabilities without TCP/IP. Truly there are things that we can do *with* TCP/IP that we can't do without TCP/IP. Many things we're already doing without TCP/IP could be done better *with* TCP/IP. That won't convince all packeteers to use TCP/IP from their homes. At this time TCP/IP still requires a computer somewhat more powerful than a TNC.

NEDA requests that TCP/IP hosts using the network access it via dedicated point to point links. TCP/IP access via 2 meter user ports is bad. They must also coordinate TheNET parameters with WZ2B. The reason is that TCP stations are themselves networking hosts. They can source high volumes of data to the network at will and can be remotely controlled from other TCP stations to do this. In order to function across the network TCP switches, by there very nature, must be independent of the controls that TheNET places on network traffic through the TheNET parameter controls. Specifically, retry rate and window size are controlled for NEDA AX.25 users but couldn't be controlled for TCP/IP users. In addition the NEDA network user access policy is designed to permit *equal* access to network services by all stations. Because of the way packet works in regards to hidden transmitter syndrome, stations transmitting into a user port cause much more loading than stations receiving *from* a user port so having TCP/IP stations on shared 2m channels would not be fair.

Hopefully in the more distant future there will be NEDA nodes that are based on NOS (Net operating system) which is a program that supports TCP/IP *and* TheNET. One or more ports on the NOS node would then be available for TCP access to the network. An intriguing capability of NOS is the ability for a TNC-only user to connect into the NOS node and then use it's TCP features.



—NEDA

CROWD: Conference Nodes

One of the TheNET software versions available allows for a roundtable or conversational type of QSO between multiple stations. NEDA has made the use of node sites equipped with this software easy to recognize by giving these special feature ports a distinctive name. NEDA converse ports are all called CROWD regardless of their callsigns or location in the network. This makes it extremely easy for keyboard users to find and access a particular CROWD port for a fun time keyboarding with other packeteers. All a user has to do is connect to any node that hosts a CROWD port and type C CROWD to be automatically connected to the local **CROWD** port.

The usage and commands for these ports is extremely easy to learn. The main commands are all proceeding by a forward slash line "/" and are entered on a line by themselves in order to be accepted as a valid command.

The commands and what they do are as follows:

/? Help

This is the HELP command. It will give you a page full of commands and what they do. You may also enter a /H to get the same thing.

/W Who

This stands for WHO and will give you a listing of who is logged onto the CROWD port and what channels they are on. There are 255 channels on each crowd and each channel can theoretically have 255 users.

/M Message

This is the command to send a *PRIVATE* message to another station who is logged onto the same channel that you are. Usage syntax is:

```
/M WA2TVE  Hi Howie, nice to see you here!
```

Where WA2TVE is the station to whom private text is sent and the line of text that follows will be the private message. The sent text will only be seen by WA2TVE.

/C Channel

In order to change channels you would use this command.

When 1st logging onto a CROWD you will be on channel zero. To change to another channel you send the slash C followed by a space and a number between 1 and 255. See the helpful hints for suggestions on gentlemen's agreements regarding channel usage.

/I Invite

If you wanted to invite someone to join you on whatever channel you are on use this command. Syntax is slash I followed by a space and the callsign of the station you want to invite. That station will get a one line blurb asking him to join you on whatever channel you sent the invite command from.

/B Bye

Or if you prefer a /Q for QUIT. This logs you off the CROWD port and disconnects your stream into the port.

One last operational point on CROWD usage is that you must send at least a <return> command after connecting to CROWD for it to actually enter you into its list of users and show you as logged on. For convenience sake the best thing to send after getting the initial connection back from the port you are accessing is the /W command. This will not only get you initialized on the CROWD but also show you just who is logged on and already there.

Helpful Hints about CROWD nodes

The following usage conventions are suggested as good operating practices when using a CROWD node in the N.E.D.A. Network. The conventions are designed to keep loading to a minimum while allowing the most effective use of any CROWD port.

1> When you have more than *three* users on a CROWD, move the group to any channel except channel zero. This makes it possible for a new user to log in and see the /H or /W command without being bombarded with traffic from your existing conversation.

2> At some times during your operation on CROWD you might find that other people's response to your comments are delayed so long that conversation becomes unfriendly. When this happens *do not* send lines of remarks regarding this phenomena. At least, don't send it in an open message. Take advantage of the /m feature when the CROWD seems to be terribly slow. Otherwise it'll just get slower.

3> Keep an eye out for *new users*, who are easily confused and haven't learned how to use the CROWD node yet. It is best if someone instructs a new user by chatting with them on channel 0 before dragging them up to an active QSO channel. This concept alone is reason enough to leave channel 0 clear most of the time when the crowd is in use.

4> <*PRIVATE*> messages can *still* be seen by other people, particularly from the various USER ports it finally comes out on. Be discrete and use operating and conversation habits that act like *everybody is listening*, as if you were operating any other amateur mode.

5> The node ops of the network nodes are responsible for making things run right. Don't be afraid to send packet mail to the sysops of a particular site to report a problem or irregularity. Use the INFO command at the user port of the site hosting the CROWD to ascertain the site operator. NEDA node ops would rather get several complaints about system problems than not know that anything is wrong.

9> Extreme "upper" channels have been suggested as places to hang out when waiting for a sched (IE: above channel 200). If it seems that stations are hiding up in the 200 range it is probably because they aren't able to pay as much attention to the conversation as they would have to if they stayed on the active channel while waiting for their sched (specific stations).

10> If you log into a CROWD, stay there for at least several minutes! Sometimes when the network is loaded, it takes several minutes for traffic to travel through the network and then another couple of moments for the station monitoring the CROWD to get back to you.

11> If a CROWD is inactive, feel free to leave a stream from your TNC on channel 0 waiting for someone else to log in while you tinker in the network on another stream from your TNC. This way someone else might check in and find you on the CROWD. This is a great way to meet people and to promote keyboard to keyboard conversation. By the way the current 'no activity time-out' on CROWD nodes in the NEDA network is 2 hours. The good news is that you can leave your terminal on in your shack and work on something else. The bad news is that if you walk away people will be saying "Fred? You there? Fred? hellloooo? Fred??" until your two hours is up!!!

12> Don't get stuck in a rut. There are CROWD nodes at many of the NEDA network sites (and some outside of our network). Try them all, what the heck, try them all at the same time! (and hope that you are well practiced at keeping your TNC streams straight!) See NEDA user port maps for CROWD sites.

13> If an emergency net starts up on the CROWD port that you are using the net control station may ask stations to either log off or move to certain channels for various purposes. Please comply as all of the capacity of the network may be required for emergency traffic. The CROWD nodes (and amateur radio) are here primarily here as an emergency resource. We just have fun with it in mean time and hope that we never have a real emergency.

Running your own CROWD

Once you have a multiport node set up with at least three user ports able to access it via backbone links a CROWD node may be added. If a CROWD node is added to a lesser network it only adds to the frustration of the users. The CROWD node is a TheNET chip similar to that used in network nodes. It runs in a TNC2 and hooks up to the diode matrix in your multiport node. The radio port is also active and can be used as a backbone port although the neither routing info or parameters are remote programmable so this is not recommended except in emergencies.

The CROWD software is available from most of the Network Regional Coordinators and is normally handled on PC compatible floppies in binary data form. The NRCs could burn an eprom for you if supply a 27256 or compatible.

Note that too many CROWDs on a network spread the users thinly to the point where they don't run into each other anymore. That is why the CANDGA CROWD is advertised as being the 'main' one. Only if the concentration of users is good do CROWD round tables generally start. CANDGA is almost always active in the evenings.

What does it look like?

Here is an example conversation on CROWD. In this conversation Doc, WB2JAB, Tadd, KA2DEW and Pete, N2IJW were already talking when KA0PGQ signed on. Tadd is recording this so it is seen from his perspective. Tadd's typing is in italics.

```
cmd:
c potsdm-5
cmd:*** CONNECTED to POTSDM-5
c crowd
POTSDM:K2CC-1} Connected to CROWD:KA2DEW-7
/w
CROWD:KA2DEW-7> POTSDM CROWD /w->who /h->help|1091
User    Circuit           Channel
WB2JAB  CANTON:WA2MZF-5   0
N2IJW   CANTON:WA2MZF-5   0
KA2DEW  POTSDM:K2CC-1     0
***
Now I'm back on.
<N2IJW>:   YUP, STILL HERE. THOUGHT ABOUT
           GOING TO GREEN N GOLD GAME, BUT
           NOT MUCH DIFFERENT THAN GOING TO
           THAT MIDNITE (12:01 PRACTICE ON
           OCT.5TH.
<WB2JAB>:  OK WELL EVEN I CAN UNDERSTAND
           THAT !
<N2IJW>:   DOC, ARE YOU DIGIPEATING THRU ME? SEEMS
           MY RIG IS XMITTING A LOT!
<WB2JAB>:  HOWS THE JOB GOING PETE ?????<<
<WB2JAB>:  NO PETE I AM TO 'CANTON' V MNY THEN
<WB2JAB>:  CROWD..???????????
<N2IJW>:   IT'S GOING WELL. FEW CHANGES SINCE I
           BEGAN, BUT ALWAYS LOOKING UP. AM SURE
           GLAD THAT I MADE THE CHANGE IN JOBS.
<N2IJW>:   OK, SOUNDS LIKE GOOD PATH.
I think that that old job sounded pretty
bad Pete. You can WALK to work now.
<N2IJW>:   YES, I WALK ALMOST EVERY DAY. EXCEPT THE
           DAYS WHEN THE WEATHER SERVICE REALLY
           BLOWS IT. THEY CALL FOR RAIN, I DRIVE,
           AND THE DAY IS SUNNY!!
<WB2JAB>:  YOU STILL GOING TO WORK DAILY PLANET AT
           THE GAMES PETE ???
<N2IJW>:   DON'T THINK SO, DOC. CAN ONLY FLY FOR
           FREE FOR THAT ONE YEAR.  ALTHOUGH I
           HAVEN'T ASKED GARY MIKEL ABOUT IT.
<WB2JAB>:  TADD HOW MANY CHANNELS IS ON CROWD ?????
<N2IJW>:   RIGHT NOW, THERE'S A DUMB SHOW ON TV...
           WILL HAVE TO CHANGE THAT!
It's got 255 channels but it can only
handle 32 users.  Pretty dumb eh?
*** KA0PGQ signed on
Howdy PGQ! Tadd here.
<N2IJW>:   HOWDY...PETE HERE.
```

<WB2JAB>: HELLO KA0PGQ HOW'S BY YOU, & WHAT'S NEW IN THE "ZOO" ??
<WB2JAB>: MY NAME IS "DOC" QTH IS WINTHROP, N.Y.
<KA0PGQ>: zoo??? the only zoo i know is the one I left in germany. Am now at Drum.
How long were you in Germany?
Also how long are you going to be at Drum?
<N2IJW>: PGQ, WHAT'S YOUR NAME? GERMANY, DID YOU LIVE THERE?
<KA0PGQ>: ok there is tod,pete,doc..... sounds like we are missing snow white
<KA0PGQ>: name here is rob
<KA0PGQ>: I will be here for a while
Tadd, you cretin!
<KA0PGQ>: and I am not shure how long I will be here
<WB2JAB>: HI ROB NICE MEETING YOU ON.
<KA0PGQ>: I wuz in da land for 3.5
<N2IJW>: ROB, DO YOU SPEAK GERMAN?
<KA0PGQ>: call there was/is da2xl
<KA0PGQ>: I was in giessen which is north of frasnkfert
<KA0PGQ>: sprect bissen doitch
<WB2JAB>: HAY TADD CAN LINDA COME TO THAT MEETING ??????<<
<N2IJW>: SEHR GUT! ICH SPRECH SEHR GUTES DEUTSCH, ABER NUHR NICH MIT SCHRIFT!!
<KA0PGQ>: ??????? cat got ur tung????
Sure thing. It's an open meeting. It'll probably be > 2hrs long tho.
<KA0PGQ>: youbetcha
We can send out for pizza if it gets too late.
We've got the room until 9PM.
<N2IJW>: ROB, MY GERMAN IS FLUENT, BUT NOT IN WRITTEN-FORM!
<WB2JAB>: BY THE WAY ROB LINDA IS 3, & WEIGHS ABOUT 110*.
3? 110?? That's hard to handle.*
<KA0PGQ>: what are you all runnin? I have a KAM and a tandy laptop PC runnin pacfile
<N2IJW>: <<<<<<<<<< AND LOVES MILKBONES!!!!!!!!!!
I'm using a Mac with a PacComm Tiny 2 and a Kenwood into a dipole on the roof.
<WB2JAB>: WELL SHE DOES HAVE A MIND OF HET OWN !
<N2IJW>: HAVE A TELEVIDEO-950 DUMB-TERMINAL, WITH A PAC-COMM TINY HERE.
<KA0PGQ>: 110 is that kilos ore carrats .
<WB2JAB>: RIGS ARE: TANDY 1000, MFJ-1278, KENWOOD TR 7730 , & A RINGO RANGER 2.
<WB2JAB>: LBS. ROB GOT SOME TO GROW YET.
Rob, you going to make the packet meeting tomorrow?
<KA0PGQ>: I have a amcomm radio ... very old fm mobile
<KA0PGQ>: how tall???
<N2IJW>: RIG HERE ALSO OLD...MIDLAND 13-500 XTAL-RIG.
<WB2JAB>: ABOUT 22", SHE HAS A BIT TO GROW YET, BUT NOT MUCH.
<KA0PGQ>: I must work tomorow ... and I have no transportation any way
<N2IJW>: TADD, MAYBE ROB CAN CATCH A RIDE WITH KA2QJO, KRISS...IF HE'S COMING.
Yup. It is too late at night to arrange that though. Sigh.
<WB2JAB>: BY THE WAY ROB LINDA IS A BLOODHOUND..
<KA0PGQ>: 22" 110 lb 3? what gives????
<KA0PGQ>: I must work tomorow I
<KA0PGQ>: rrr woooffff
<WB2JAB>: ITS STANDARD SIZE FOR A BLOODHOUND ROB..
WOOOOOF!
WOOF hahahaha. Funny joke Doc
<N2IJW>: ROB, SEE IF YOU CAN REACH KRISS (KA2QJO) ON THE 147.255+ REPEATER, IF YOU CAN MAKE IT...AND HOPEFULLY KRISS IS COMING.
<N2IJW>: TADD, SEE ROB KNOWS WOOF!
<KA0PGQ>: I like cats
<KA0PGQ>: I wont be able to do anything tomorow I have 24 hr phone watch
<KA0PGQ>: here kitty kitty kitty
<WB2JAB>: WELL LINDA HAS BEEN AT K2CC BEFORE TADD.
<N2IJW>: TADD, LINDA MIGHT BE ABLE TO ACT AS A GUARD-DOG....
Might?
110 and you say MIGHT?*
AAAAAAAAAAAAAAAAAH
/w

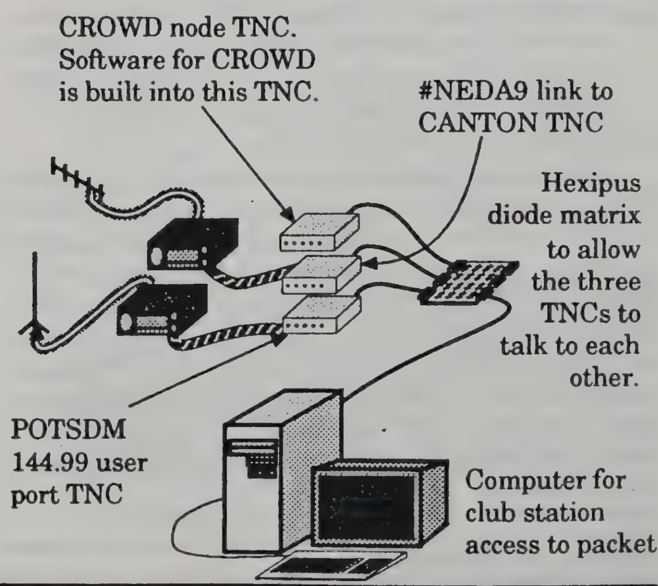
User	Circuit	Channel
WB2JAB	CANTON:WA2MZF-5	0
N2IJW	CANTON:WA2MZF-5	0
KA2DEW	POTSDM:K2CC-1	0
KA0PGQ	WATERT:KA2QJO-5	0

***

Well enough of that. Conversation on the CROWD that night ranged from talk about race cars, to guns, saxiphones, radios, packet, dogs, food, you name it and the conversation lasted for about 3 hours. It's a lot of fun. Join the CROWD.. If nobody is on when you get on just hang around. Leave CROWD on when you are working around the shack. If you do that every evening and anybody else does too they you've started a CROWD! Here's a copy of the help file from CROWD:

NEDA *CROWD* port	Commands are:
/?	Print help information
/BYE	Terminate the convers session
/CHANNEL n	Switch to channel n
/EXIT	Terminate the convers session
/HELP	Print help information
/INVITE user	Invite user to join your channel
/MSG user text	Send a private message to user
/QUIT	Terminate the convers session
/WHO	List all users and their channel numbers
/WRITE user text	Send a private message to user

***



Programming: Bill Beech, NJ7P.
Initial idea and format by Jack Taylor, N7OO.
Some text by Jack Taylor, N700.

TheNET Plus
Node Op Resource Manual

This version of the resource manual by Tadd Torborg, KA2DEW for v2.08B of TheNET Plus

TheNET and TheNET Plus
Portable. Compatible.
Public Domain.
NORD><LINK

TheNET software is public domain, ONLY for non commercial use.

TheNET software by Hans Georg Giese DF2AU & NORD><LINK
and by William Beech, NJ7P with Jack Taylor, N7OO

The authors deny any responsability
for the product or it's use.

Table Of Contents

TheNET Plus Versions

v2.00 was the prototype test node. The maximum number of calls listed in the Heard list was 10 over a 10 minute period.

v2.01 was the first "official release" of TheNET Plus. The Heard list was changed to a maximum of 20 calls listed over a 15 minute period. A parameter was added which allowed the NodeOp to set the maximum number in the Heard list to a value less than 20, if desired.

v2.02 was not released.

<< This information copied from a document by N7OO >>

v2.03 was identical to v2.01 with the exception of a bug fix associated with Parameter 24. In v2.01 if callsign verification was turned off, it also disabled the N * function. v2.03 corrects this situation.

v2.04 corrected a problem caused by the fix in version 2.03 and changed Parameters 28, 29, and 30 to agree with the SETPLUS utility and the current documentation.

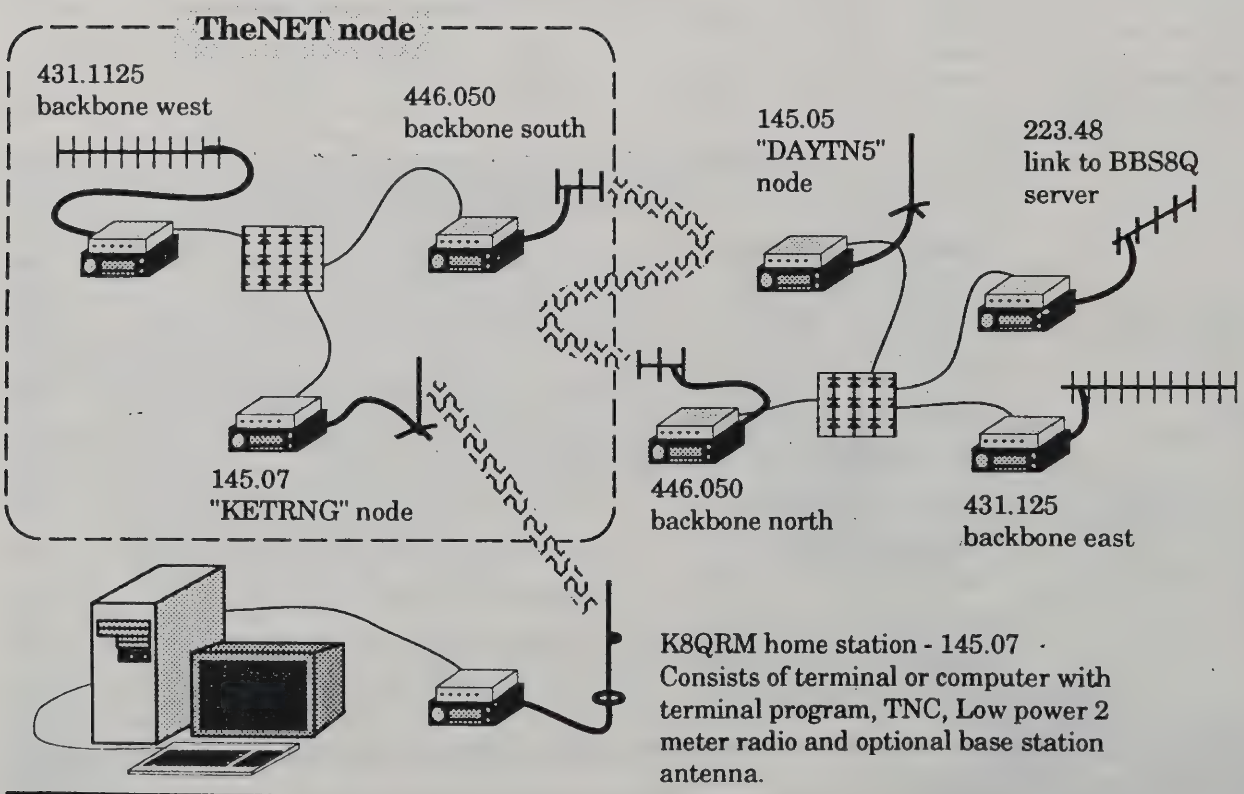
v2.05 added several new NodeOp convenience features. One was to have the STA LED light when someone connected to the node. It was felt this would assist the NodeOp in servicing his equipment while at the site. Another new feature was to add three sysop KEY commands, MARK, SPACE and DIDDLE which keys the transmitter and turns on appropriate alignment tones. Also added was an ON - OFF remote control capability. One of the NODE command responses was changed to **Host busy**, instead of **Host table full** when a user attempted to connect to the host (a non-allowable function when the host port is active). The **Not Found** response to an unknown node now indicates **Not Found: <node alias>** to assist those running multiple streams in identifying which stream is which.

v2.06 corrects a long standing routing display anomaly, which takes care of the last known problem.

v2.07 was not released.

v2.08 adds connect command disable, #node propagate disable option, adds broadcast via port options and reorders parm list.

v2.08B makes ROUTES command response show both mnemonic and callsign.



Forward

TheNET is a good choice for network building. It has several key qualities which make it a very good choice for an amateur radio packet network building block.

- It allows for multiport nodes;
- Backbone routes are easily protected against miscreant abuse while still being monitorable by any ham with a scanner and TNC (and appropriate modem)
- It supports DxCluster, TCP/IP, keyboard to keyboard, BBS forwarding;
- It can be implemented in a harsh environment;
- RF tightness of all digital hardware is easy to assure
- It's software is stable and mostly bug free;
- Network architecture is verifiable remotely by all users;
- Operating parameters are visible to all users;
- It's simple;
- It's cheap (less than $130 per port for digital hardware);
- It's incrementally upgradable;
- It's fun for the average user;
- It's compatible with CROWD conference nodes which may be the single most affective packet recruitment device.

TheNET has been targeted as a poor network building block in the past. This criticism has most probably been based on experience with a poorly designed network. It is important, in ham radio packet network implementation, that good design practice is followed. This manual has information which is required to build a successful TheNET based network.

Background

NORD><LINK in Germany created TheNET with the first release called TheNET Version 1.0. The software was distributed in both EPROM image binary form and source code. The software was public domain so other software people could join in the project.

Shortly after TheNET arrived on the scene most digipeaters were upgraded to be single port TheNET nodes. This was deemed desirable because of the *store and forward with local retry* feature of TheNET. At that time there were very few packeteers and scant services. Most hams that used packet at all used it to connect to a local BBS and acquire or leave traffic. The BBSs performed all of the higher level functions. Users could only access BBSs that were relatively close geographically. Performing complex routing operations for the purpose of accessing remote servers was frowned upon. To the average ham it looked like packet was no fun at all unless you were one of the BBS ops. In some of the metro areas many BBSs were brought on line by the packeteers in search of 'fun'. Because there were very few frequencies in use, and with cross frequency connections rare, long distance connection to a BBS was discouraged. There were ego-wars about who could put up a BBS in a certain area or on a certain frequency.

Because the only implementations of TheNET were single port (just modified digipeaters) and because packeteers in general were discouraged from adding to the existing functional services base, packet growth stagnated. Basically the only way to develop new hardware was to create the users group first, and then start working on hardware. Many users groups were founded on the proposed existence of hardware or software. These groups never lasted long and once again the average packeteer was abused.

Packet radio used VHF and UHF spectrum This is the best place for automated packet stations and user access to BBSs because communications on VHF and UHF is more consistent than HF and there is much more spectrum available. The best people to work on a packet radio 'network' that already had experience on VHF and UHF are the repeater and FM enthusiasts. Strangely, in the early days of packet radio (1980 thru 1988) the repeater and FM enthusiasts not only didn't want to work on packet systems but they despised the mere concept of packet. They didn't like packet taking up FM simplex channels and repeater spectrum and they didn't like the fact that they couldn't monitor the packet traffic by ear. They also didn't realize the potential of packet ra-

dio for long distance shared channel communications, an implied packet radio feature that still didn't exist anywhere. So the most valuable resource available to ham radio packet had declared packet radio networking a non-goal, at least for a while.

Now, back to the BBS ops. After operating a BBS for a short time many of those who came in search of fun found out that it was actually a lot of work. Some eventually dropped out of packet radio altogether. The remaining BBS ops have generally seen packet radio decrease in usefulness. Some have banded together to create better facilities for trafficing their BBS data.

TCP/IP has seen some popularity, starting in the mid 80s. TCP/IP offers real time and non real time operation modes. Most of the stations involved in that branch of the hobby are pure computer science enthusiasts. They are in search of amateur radio links to extend the usefulness of their computers. Most do not have a strong background in radio, even less than the BBS ops. TCP/IP suffers from a lack of packet connectivity as badly as the BBS operations do.

DxClusters are a creation of AK1A and the Yankee Clipper Contest Club (YCCC). This is a product that, like a BBS, runs on a PC. Unlike a BBS it establishes communication between all available DxClusters and attempts to keep the connection up. Users who connect to the local DxCluster have access to the stations and facilities at all of the other DxClusters. The only flaw with a DxCluster network is that it only supports DxCluster features. This leaves no room for TCP/IP, DOSgates or compatibility with TheNET, ROSE etc.. DxClusters can talk over a TheNET or ROSE network but ROSE and TheNET can't talk over a DxCluster network. No effort is made to preserve 2m channels as this seems to be very under stressed in the DxCluster literature.

In the middle of all this are the hams that got onto packet for the sake of communicating. These people are interested in performing similar operations on packet that they would via voice modes. They want to go out and find people to chat with. They'd be happy meeting the old crowd or finding new hams to talk with. These people have been totally frustrated with packet radio. Perhaps three quarters of all hams who got into packet radio fell into this category. Many have forsaken packet radio. Some have decided to do something about it. In the north east United States a club was formed of ham radio packet communicators. This book is the product of their experimentation and successes.

TheNET has much more potential than it has been used for. TheNET allows a network to be constructed that is much more efficient than a single port node network. In order to take advantage of the true power of TheNET (or any other of the currently existing networking software) a proper radio system needs to be established. Carrier Sense Multiple Access (CSMA) is the current mode of packet operation in use by hams today. CSMA *will not work* on a system where there are hidden transmitters. In order for a packet radio network to function all links to servers and all links between nodes must be dedicated point to point links. Only in this fashion can an environment be created that allows for expandability and upgrade. Without this inherent throughput, expandability and upgrade path no network will be successful in the short run, let *alone* the long run (unless all contributors are very rich to start with and infinitely high speed equipment is used at the start). Furthermore the equipment recommended to potential subscribers must be available off the shelf. If a limitation on network subscribers is created by requiring them to be software or RF gurus then the network that is created will necessarily exclude those without the required skills. A TheNET node may be constructed with entirely off the shelf gear. Almost all of the gear can be found in any of the common ham radio dealerships. The remaining gear (diode matrix boards and EPROMs) is available from several packet groups and is of the smallest hindrance to 'getting started'.

One of the design criteria that went into TheNET is that any packet user on the network is privileged to look at the network architecture and to examine a lot of the network functionality. The network may also be monitored with commonly available equipment (except for high speed modems). This is a feature that allows new network subscribers (node owners) to come up to speed quickly. Without this inherent user freedom a lot of potential network builders might be turned away, mystified or feel left out.

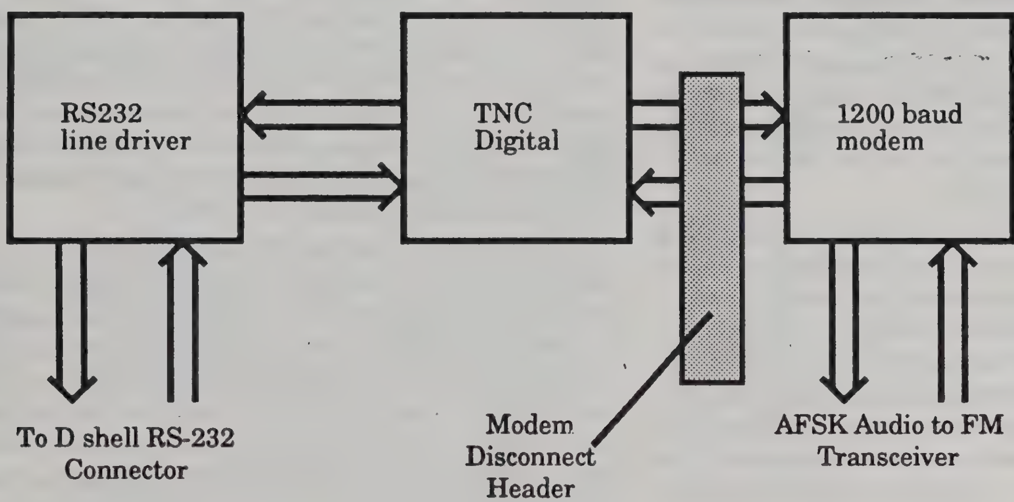
Recently NJ7P, Bill Beech, and N7OO, Jack Taylor designed and implemented changes to TheNET to make TheNET Plus v2.08B. Jack and Bill's purpose was to add functionality to give even more support for the network users. They have succeeded. This document describes most of the features and much of the functionality of TheNET Plus v2.08B. Keep up the good work Bill and Jack!

Theory of Operation

TheNET is a software package that runs in a TNC2. The purpose of the software is to control a TNC in a system of TNCs called a network. The network is usable by hams running any mode of amateur radio packet based on AX.25 protocol. The hams can utilize the network to chat in real time, access remote computers, pass traffic or perform paging and remote control. Follows is a technical description of the TheNET software.

Hardware - L1

The TNC2 is a dual port device. That is, it has two serial i/o channels. One of these i/o channels is hooked to an RS-232 driver and a D shell connector. The other i/o channel runs to the *modem disconnect header* and then to a 1200 baud modem.



The software that runs in the TNC2 is installed in a 32K EPROM and is mostly compiled C code. Some small sections are¡ written in assembly language. Also stored in the EPROM are default parameters and text strings. Generally the text strings are not user programmable but a bit hacker could find them and change them. Text strings that exist include "`Connected to`" message, command names, "USERS" response string, beacon text and error messages. The default parameters, callsign, node name and password are programmable. Most installers of TheNET Plus 2.08B will be using the SET208.EXE program with a PC.

International Systems Organization (ISO)

The function of a TheNET node is to act as an active data *store and forward* device as well as a remote *command interpreter* for users. Communications can occur both on the modem port and on the RS-232 port at the same time. Communications is AX.25 with networking and routing operations which are written within the bounds of the ISO 7 level model. That is that the processes in the TheNET software are modularized in the following functions:

L1:I/O control and hardware;
L2:AX.25 linking;
L3:network routing;
L4:transport processor; and
L7:command processing.

Link Controller - L2

The TheNET node's link controller will accept and make AX.25 connects on either the modem or RS-232 ports. If a station connects to the TheNET node on either port the node will remember that a connection is made, the callsign of the connecting station, and the callsign that was used to connect to the node. These are saved as the address field. The node can accept the connect using the preset callsign and ssid in the EPROM or using the nodename with any of 16 SSIDs. Connects may be accepted by the node from the same callsign on all 17 *callsign - nodename - ssid* combinations at the same time. The next time a packet is received that matches that address field the node will classify the connecting station as either a user or as another node.

If the connection is a user then the user is added to the users list and any further communications is passed to the command processor. The user may interrogate the node for information that it has (see user commands) or he may command the node using the sysop commands or using CONNECT or CQ. If the user uses the CONNECT command he may establish a connection to another node or to a user from this node. This is covered later under "*Circuit*".

Routing Processor - L3

If the connection is another node then the next message that follows will contain TheNETese. TheNETese is a slang term that means that the communications has non printable characters that TheNETs understand. More on that will be covered under *Protocol*. If the node's link controller gets the TheNETese then it marks the station as a neighbor TheNET node and passes the connection information up to the routing controller. If the traffic that is received is destined to another node then the routing processor passes it back to the link

controller to go out to the next neighbor node in the chain. A neighbor node is a node that this node can talk to directly, either over the RS-232 port or over the modem port.

Transport Processor - L4

If a neighbor node passes traffic to me (I'm a TheNET TNC) that is destined for me then my routing processor passes the message to my transport processor. My transport processor is responsible for making sure that all data that originates here or is destined to me makes its way across the entire path between circuited nodes. So, if I connect from here to another node that is many sites away it is my transport processor that is responsible for seeing that the message gets there.

The transport processor gets messages from the network processor and from the command processor. The command processor is hooked to the user. Users can connect to a node and then tell it's command processor to connect to another node. Users can tell the command processor to connect to another user or server station.

Command Processor - L7

The command processor may be instructed with ASCII text commands from a user station. Much of the remainder of this manual deals with command processor functionality. The important functions needed for understanding of the remainder of *"Theory of Operations"* is that the command processor allows the user to connect to other nodes via the network over either the modem port or the RS-232 port and to stations that are not nodes over the modem port. In addition the user can request lists of:

all nodes in the routing table;
neighbor nodes;
the best three neighbor nodes for a particular node;
all L2 connected stations known to be users.

Program Start-up

The program starts up when power is applied. It lights the STA and CON LEDs for a second and then turns them off. It initializes its memory, copying default parameters unless it has what it thinks are valid parameters and INFO message in RAM already. Then it sends a beacon message on both the modem and RS-232 ports. The node broadcast timer is started.

Routing

When the routing processor gets a packet to send out it looks at the destination address provided by the transport processor. The destination address is the callsign of the requested destination node. The routing processor looks up the node in it's node database (routing table). It will find up to three neighbor node callsigns which are in the direction of the destination. These neighbor nodes are routes to the requested node. If the requested node is unknown the message is thrown out. The information supplied with each route is the callsign of the neighbor node, the port number of the route, the quality value associated with the path and a flag indicating that a route is already in use. If no flag is set then the router selects the highest quality route and sets its flag. The port number describes whether it's an over the radio shot or an over the RS-232 shot. This information is passed to the link manager.

Slime Trails

The router knows of up to 100 nodes (adjustable) and knows of up to 3 routes per node. If a message, whose origin node is not in the routing table, is passed to the router, the router notes what neighbor node and port sourced the message and installs the origin node and route into the routing table. This way when an answer to that message comes back through the node the node will know what to do with it. This function is called slime trailing and only happens in the event that the origin node knows of destinations within the network, and where the network doesn't know of the origin node. *Important: If the routing table already has 100 nodes in it then a slime trail cannot be added.*

The reason that this function is called a slime trail is that when a user requests a copy of the routing table (nodes list) the slime trail shows up on the nodes list with just the callsign. If the user traces down the origin node by using the N command with the callsign as the parameter the user will step through nodes until he reaches the origin. At each step there will be a node listed with just the callsign. At each node along the route the slime trail route will time out randomly based on internal TNC timers.

Nodes Broadcasts

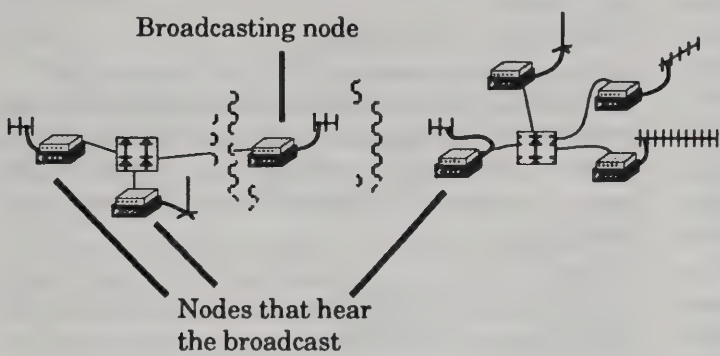
Every 30 minutes (parm adjustable) the node will send a nodes broadcast via both it's RS-232 and modem ports. This broadcast allows the neighbor nodes to maintain their databases of nodes that this node is sourcing. The broadcast consists of all of the nodes whose obsolescence counts are equal or greater than parameter 5 "Obsolescence counter, minimum for broadcast" and all of the nodes whose obsolescence counts are 0 (locked nodes). The format and order of the nodes broadcast is basically:

```
This node, PQ=256
knownnode, PQ=nodequal
knownnode, PQ=nodequal
knownnode, PQ=nodequal
...
```

nodequal is the highest of the 3 values that the node has stored for knownnode. *An interesting quirk of TheNET is that if the sysop has locked the node itself in that the node broadcast will include it as a knownnode. Because TheNET nodes believe everything that they hear in the broadcast, in the order that they hear it, the node*

can convince other nodes that it is of poorer quality than they have set for the route.

When a node receives the nodes broadcast in goes



through the following program sequence:

```
Check if route to the neighbor we
are hearing is in place.  No? Then
use parameter 2 or 3 depending on
if this is RS-232 or HDLC and put
the neighbor node in the route
list.
Apply the route quality for the
neighbor we are hearing, to the
received nodes broadcast.
As each node is clocked in store
it in the routing table under the
node's callsign with the route
indicating the neighbor we are
hearing.
As each route entry is being
stored in the routing table, set
the obsolescence counter to the
initialization value in the
parameters.  If the route to the
node we're storing is locked in,
then ignore the incoming
information.
```

Circuits

The process of connecting into a network node, across the network, and then out of the network requires three operations. The first is called an uplink. The second is called a circuit and the third is called a downlink. The uplink and downlink stages are AX.25 connections to the link processor of each of the two end nodes. The link processors pipe directly to the command processors. Circuiting means that the command processors pass the traffic to the transport processor which then passes the traffic to the router etc..

After the router in the origin node gets traffic to go to some distant destination node the traffic can hop from node to node over more than a dozen TNCs before again reaching a transport processor which will be at the destination node. The destination transport processor will then acknowledge the packet and the process is repeated in reverse. At the same time the destination transport processor acknowledges the message it sends the message to the destination command processor. If the origin transport processor hasn't gotten an acknowledgment before a time-out timer expires (transport time-out) it can resend the message. If the origin transport processor gets another message from the origin command processor it can send that one into the system as well. It can have up to 4 messages in transit without having acknowledgments. When the command processor attempts to send the 5th message the transport processor will respond with a 'choke' flag.

Using a TheNET Network

There are two common modes of using a TheNET network. These are broadly called level 2 access and level 4 access. Most users will only be concerned with level 2 access. The level number refers to the ISO networking model described in *Theory of Operation*.

Level 2 Access

The user connects from his TNC to the nearest, but least busy, network node. This is done using the connect command from the user TNC. The node may be addressed using either the node's callsign and specific ssid or using the node's mnemonic and any of the 16 SSIDs. Let's assume that our user is N2MGI and the local node is POTSDM:K2CC-1. The following are all valid connect commands:

```
c POTSDM
c K2CC-1
c POTSDM-4
```

The POTSDM node will answer any of those. All are perfectly reasonable ways to connect to the node. The reason that the node allows this versatility is that if the user can do multiple streams he can connect to the same node up to 17 times, or however many streams the TNC allows if less than 17.

In this example the node would answer and a *** **Connected to** message would show on N2MGI's screen. A monitoring station would observe:

```
N2MGI>POTSDM (c)
POTSDM>N2MGI (ua)
```

or

```
N2MGI>K2CC-1 (c)
K2CC-1>N2MGI (ua)
```

or

```
N2MGI>POTSDM-4 (c)
POTSDM-4>N2MGI (ua)
```

The node will assume any of the 17 identities for the purpose of maintaining the connection. N2MGI could, on three different streams, connect to all three of these identifiers.

Level 2 Network Use

After the user has gained access to the node he can request a list of destination network sites or he can issue the Connect command to select one. Then he can use the Connect command from the destination node site to connect out of the network. The following is an example procedure where N2MGI might connect to KA2DEW using the network. From the command prompt N2MGI types:

```
cmd:C POTSDM
```

Shortly his TNC answers:

```
*** Connected to POTSDM
```

N2MGI is now connected to the TheNET node POTSDM. There are several commands available at this time. (See User Command List). For MGI's purposes of connecting to KA2DEW from CHSTR node he then types:

```
C CHSTR
```

This tells the POTSDM node to pass MGI off to the CHSTR node. POTSDM returns:

```
POTSDM:K2CC-1} Connected to CHSTR:K1MEA-2
```

Once again MGI can enter any of the TheNET user commands. To get to KA2DEW he types:

```
C KA2DEW
```

POTSDM node receives the "C KA2DEW" from MGI and passes it off to CHSTR. CHSTR makes the connection to KA2DEW and then sends it's response back through the network to POTSDM which then sends:

```
CHSTR:K1MEA-2} Connected to KA2DEW
```

KA2DEW's station, which has gotten a connection from CHSTR under the callsign of N2MGI-15 responds to CHSTR with a connect text. That text is sent through the network to POTSDM which then sends it to N2MGI. So:

```
Tadd's station.  Use KA2DEW-4 for my PMS
or beep several times to get my attention!!
```

Now any traffic that N2MGI or KA2DEW type will be routed back through the network. The network is now transparent to the two stations.

Level 4 Access

Some equipment that a user might operate or that is built into certain server systems is capable of directly accessing TheNET through the higher levels of the ISO model. In this case the TheNET node would be accessed such that it thinks that the user/server is yet another TheNET node. The user/server could access the command processor at the local or at a distant TheNET node or might perform level 4 access direct to another user/server across several nodes.

User Command List

Once a station makes a connection to a node everything sent into the network will be handled by the node's command processor. These are the commands available to the user.

BYE

This command will tell the node to disconnect from the user and may be abbreviated to B. This will have a similar effect to the user doing a disconnect from his own station.

CONNECT

This command instructs the node to connect to another station. The command is entered as `C STATION` where station may be a six character or less text string or a valid amateur callsign. First the node searches it's own database for a match between *station* and a known node name or a callsign associated with a known node. If a node name match is found then the callsign associated with that node is used. If that is the case or if *station* matches a node's callsign then a network connect is attempted to the requested node.

If no match is found then the node will process STATION and determine if it is a valid callsign. If not then the node will send an error message to the user. If it was a valid callsign then a connect attempt is made via the modem port of the node. If successful the user will be sent `nodecall:nodename} Connected to STATION`. If unsuccessful the user will be sent an error message.

CQ

The CQ command is used by a station who wants to make a random connection from a node. The node may either be the local node, or a remote node. The CQ command is sent with a parameter of up to 77 characters. The node will send a message with the calling station's callsign (-15), to CQ, with the parameter text in the info frame. Thus if user NK1M types:

```
cq Bill calling from Nashua
```

The node will send over the air:

```
NK1M-15>CQ: Bill calling from Nashua
```

The node only transmits the text once. If Bill wanted the text sent several times he'd have to type it several times. The node puts NK1M into CQ mode. That means that if it hears anybody trying to connect to NK1M-15 it will complete the patch, connecting the new station back through the network to NK1M. Additionally while NK1M is in CQ mode the USERS list will show NK1M-15 as calling CQ:

```
Circuit (MONROE:WB2GNR-1 NK1M)  <--> CQ (NK1M-15)
```

If a station connects to the node that NK1M is calling from and then connects to NK1M-15 from the node, the node will make the patch. CQ mode times out and disconnects NK1M after "No-activity timer" runs out (usually 2 hours). See *Node Parameters*.

HEARD

The H command requests a list of stations that the node has heard in the past 15 minutes. Stations that are known to be nodes are ignored. The maximum number of listed stations is 20. The stations are listed in an odd order, not by time. Stations that digipeat before they are heard by the node are shown anyway, but neither the digipeater, nor the fact that they were digipeated is indicated.

INFO

Sending this command will make the node respond with a block of text that will describe the node's location, frequency, who to contact, servers accessible etcetera. Examples of info messages are:

```
WB2QBQ-1:KNOX}
port  144.91 USER
QTH   Knox, N.Y.
sysop WB2QBQ @ WA2PVV
phone 518-555-1212
maps  NEDA Box 563 Manchester NH 03105


K2TR-10:DXKNOX}
Port  Dedicated DxCluster Link
QTH   Knox, N.Y.
info  WB2QBQ @ WA2PVV
Enter C K2TR for connect to YCCC/AARA DxCluster
System


N2CJ-1:CLV}
Port  145.09 MHz USER access channel
QTH   Clove Mt. Poughkeepsie, NY
spnsr N2CJ @ WB2COY
servers pls use 441.0 for network access!

For server info C BBSCOY  DN CLVSERV.TXT
```

NODES

This command gives the user access to the routing table in each node. After connecting to a node a user can use `N`, `N *` and `N <alias or callsign>` to get information from the node about it's routing table.

The `NODES` command (abbreviated `N`) returns a listing of the user port nodes contained in the routing table. It gives a user a listing of possible destination nodes for him to connect to.

```
N
SRTGA1:WA2UMX-1} Nodes:
WA2PVV              WA2UMX-4            BBSUMX:WA2UMX       GFL:N2AYY-1
KINDER:WA2PVV-7     NEDALB:WA2WNI-3     OTSEGO:W2SEU-1      SCROWD:WA2UMX-7
SRTGA2:WA2UMX-2     SRTGA5:WA2UMX-5     SRTOGA:WA2UMX-14    WMA220:K1FFK-2
```

The listings which include a six character (or less) mnemonic followed by a callsign are nodes whose information was received via a node routing broadcast. The listings which include just a callsigns are nodes whose information was received in order to create a return path to a otherwise unknown node. This function is called *Slime Trailing* and is described in the *Theory of Operations*.

The `NODES *` command (abbreviated `N *`) returns a listing of all of the nodes contained in the routing table. This includes the # nodes. (See *Selecting Mnemonics*)

```
N *
SRTGA1:WA2UMX-1} Nodes:
WA2PVV              WA2UMX-4            #GFL10:N2AYY-10     #SCR10:WA2UMX-10
#SCR11:WA2UMX-11    BBSUMX:WA2UMX       GFL:N2AYY-1         KINDER:WA2PVV-7
NEDALB:WA2WNI-3     OTSEGO:W2SEU-1      SCROWD:WA2UMX-7     SRTGA2:WA2UMX-2
SRTGA5:WA2UMX-5     SRTOGA:WA2UMX-4     WMA220:K1FFK-2
```

The N command can be used to determine the neighbor node and quality for a particular node. The syntax is `N <alias or callsign>`. An example:

We are at the CANTON node and wish to know the route to POTSDM. We issue a `N POTSDM` command and receive back this response:

```
CANTON:WA2MZF-5} Routes to: POTSDM:K2CC-1
> 128 3 1 #NEDA:WA2MZF-11
   81 3 1 #NEDA:WA2MZF-10
```

This tells us there is a route to POTSDM and it is an RS-232 path (the 1) via WA2MZF-11 which is a backbone node and this route is currently in use. The numbers given in the N <alias or call> command will be explained later. Here we just want to show how the N <alias or call> command is a powerful tool to help one navigate throughout the network.

PARMS

(Parameters) Issuing this command will yield a status listing of the nodes parameters. There are 33 parameters although only 16 are answered with by the PARMS command. The other 17 are only visible during node EPROM setup.

This is a serious bug as in this version of TheNET the users can't access ALL of the node's operating parameters. Hopefully this will be fixed in later versions

The node response may look like this:

```
POTSDM:K2CC-1} 50 0 203 3 4 1800 4 1 10 0 1 0 35 2 0 1
```

The convention is to number the parameters from left to right so parameter #3 is a 230. Each parameter affects the node operation in one way or another. See the section titled *Node Parameters* for a complete description of the parameters.

ROUTES

This command yields a listing of all radio line of sight or wire connected nodes that are directly worked (at level 2) by the node. These nodes are called *neighbors*. The listing will also show nodes and digi routes set by the sysop locking commands. Due to the different protocols involved, TheNET does not recognize KA-Nodes, ROSE nodes, or TEXNET nodes in it's routes list. It will recognize G8BPQ, MSYS and compatible TCP/IP nodes. A typical routes display may look like this:

```
CANTON:WA2MZF-5} Routes:
  1 #NEDA:WA2MZF-12 203 16
  1 #NEDA:WA2MZF-10 203 3
> 1 #NEDA:WA2MZF-11 203 12
  0 HULL:VE2RBH 50 1
```

In column one we see a 1 for all paths that are through the *matrix* and a 0 indicating a *radio* path to VE2RBH, HULL node. The right arrow indicator tells us one of the paths is either in use or has had activity within the past 15 minutes. All radio paths show a standard path quality value of 50 (This is a standard *user* port). All RS-232 paths show a path quality value of 203. The last column indicates the number of nodes sourced from this route.

Sysop Command List

In addition to the **user** command listing given above there are a special set of commands for System Operator use. To be able to use these, you will have to be recognized by the node as a sysop. The method for doing this is described below in the SYSOP command. It is also possible to use these commands through the host mode connect port. See *Host Interface*.

SYSOP

After connecting to the node by issuing a "C callsign" command, type "S". S will return you a random series of five numbers separated by spaces. Each number refers to a single character in the password string. 1 refers to the first character. 22 refers to the 22nd character. All you do is type the five characters indicated and then hit <return>. The node will *not* tell you that you have been successful. This would be too obvious to a listening station. You can actually run the SYSOP command several times, correctly answering only one password and falsely answering the rest. A listening station won't be able to figure out which one you answered correctly.

To test the success of your SYSOP command, type **P**. This will give you a string of numbers, representing the default values for the various node parameters. Note the value of the first number (typically 50). Now type **P 51**. If successful, the first parameter should have the new value 51. Again type **P** space and insert the original number back in the parameter listing (**P 50**).

Sample password string:

```
FRED WAS A BIG HERO AROUND HERE UNTIL TH
```

I usually write out my password strings in a matrix so they are easy to translate:

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9
0x	!!	F	R	E	D		W	A	S	
1x	A		B	I	G		H	E	R	O
2x		A	R	O	U	N	D		H	E
3x	R	E		U	N	T	I	L		T
4x	H									

Remember that the first valid digit is 01. So, the following exchange would properly sysop the LYNNWD node:

```
SYSOP
K7QRM:LYNNWD} 31 13 40 01 08
```

The 31 indicates the 31st character in the password string. From the matrix we get the second character from the 3x line. The first character is character 30 and is a R. The second character is character 31 and is an E. Next we get character 13 from the 1x line which is an I. Continue for all five characters. Note that the node will not return a number that represents a space character. So, I type:

```
EIHFS
```

The node won't respond to this so in order to verify that I got in I type:

```
P
K7QRM-1:LYNNWD}  50 50 230 3 4 1800 6 1 8 0 1 0 35 2 0 1
P 51
K7QRM-1:LYNNWD}  51 50 230 3 4 1800 6 1 8 0 1 0 35 2 0 1
P 50
K7QRM-1:LYNNWD} 50 50 230 3 4 1800 6 1 8 0 1 0 35 2 0 1
```

Note that the first parameter changes from 50 to 51 and back to 50. It doesn't change then you have the wrong password or you didn't respond correctly. It is very important that the parameter is changed back! So, if you do this procedure don't change the first parameter by too much. If you were to change it to 255 and then weren't able to change it back (Your tower just blew down) the node would soon become useless.

A suggested alternate way to record password strings is with the character numbers on the line above the password. This lets the sysop record several passwords on one page. Example:

```
0    5    10   15   20   25   30   35   40
|    |    |    |    |    |    |    |    |
 FRED WAS A BIG HERO AROUND HERE UNTIL TH
```

INFO

Allows the sysop to enter text into the soft coded INFO section on the node. This INFO section has available a maximum of 160 characters. Up to 80 characters on one line is allowed. It is always a good idea to start any info string with a blank line feed to allow your info text message to be formatted nicely on the user's screen. This is done by entering a control A on the first line as shown below. The reason a control A is used is that a control A is an innocuous character that will invisible to most users. Note that a "I" followed by just a ctrl A will blank the entire info message. For text to be entered on the 2nd and subsequent lines, type: `I +<text>` until the total 160 character limit is reached.

As this technique is different than previously used, a little in-house practice is advised until you become familiar with it. Review the INFO format examples previously given for ideas on how you want to set the INFO text format. An example for setting INFO:

Sysop action: `I ctrl A  <return>` (notice, no + sign)

Node response:

`ALBANY:WA2WNI-1}`

Sysop action: `I +port  144.93 USER`

Node response:

```
ALBANY:WA2WNI-1}
port  144.93 USER
```

Sysop action: I +cnty Rensselaer

Node response:

```
ALBANY:WA2WNI-1}
port  144.93 USER
cnty  Rensselaer
```

Sysop action: `I +QTH   West Grafton, N.Y.`

Node response:

```
ALBANY:WA2WNI-1}
port  144.93 USER
cnty  Rensselaer
QTH   West Grafton, N.Y.
```

Sysop action: `I +info  WA2WNI @ WA2PVV`

Node response:

```
ALBANY:WA2WNI-1}
port  144.93 USER
cnty  Rensselaer
QTH   West Grafton, N.Y.
info  WA2WNI @ WA2PVV
```

This process can be repeated until the maximum of 160 characters, including non-typing and punctuation, has been reached.

The key things to tell your connecting stations are:

- What frequency the port is on;
- What type of port it is;
- Where it is located;
- How further information can be obtained about the network or the club;
- Callsign and BBS for the sysop and/or sponsor of the node.

If all of this information is available then someone in your area who wants to link to your node can easily get in touch with you. That way network growth can be facilitated.

KEY

`KEY MARK`; `KEY SPACE` or `KEY DIDDLE`

Operates the PTT line of the TNC and turns on the Mark (high) tone, the Space (low) tone or a diddling between the two tones for approximately 22 seconds.

The purpose of the key commands are to make the NodeOp's job of setting deviation and RF frequency much easier. Previously it was necessary to reinstall the original TNC firmware chip and perform the CALIBRA procedure in order to set FM deviation. Now if the node-op has the appropriate equipment and is within radio range of the node, he can routinely check frequency and deviation remotely! At the site, this same procedure can be done via the host mode interconnect. Once entered, there is no way the KEY command can be terminated until the internal timer runs it's course. If the node watchdog timer is set for less than a 22 second duration, the watchdog will unkey the PTT. However, the node will continue the KEY command until the remaining time has expired. During this period, the node will not execute any other commands.

ON - OFF

A simplified remote control capability as compared with the HIGH LOW commands found in TheNET version 1.0. *ON* turns on the CON LED and *OFF* turns the LED off. In the MFJ 1270B, the voltage sense from the CON LED appears on pin 8 of the DB25 connector. The voltage sense also appears on the base of Q16, a 2N3904, the output of which goes to pin 2 of the TTL connector. The main caution here is that the control switch be capable of passing the appropriate amount of current and voltage to the controlled device.

PARMS

Allows the sysop to make changes to parameters 1 through 16 over the air. These changes are permanent until a RESET command is performed or until battery power is removed from the TNC's RAM. In order to change parameters 17 through 33 a new EPROM must be burned.

To use, type `P ************** 650 <enter>` to change parameter 15 from 900 to 650, for instance. The asterisks preceding the value to be changed protects the preceding parameters from being changed. To change Parameter 4, type

`P *** 245.`

For more parameter information see *Node Parameters.*

NODES

Occasionally a need may arise to modify node entries in the node routing table. The sysop command for this is:

```
N <call> + <alias> <quality> <obs> <port>
<neighbor> (digicall1, digicall2)
```

For example:

```
N WA2TVE-8 + DXCLUS 58 0 1 WA2WNI-7
```

In this example the node DXCLUS:WA2TVE-8 is added to the nodes list with the node name of DXCLUS with a route quality of 58, and obsolescence of 0 (thus locking in the node) via the RS-232 port (the 1) and routed to WA2WNI-7 as the neighbor. Setting the obsolescence to a non zero value will cause this planted node information to be temporary

To manually unlock DXCLUS, the command is reversed:

```
N WA2TVE-8 - DXCLUS 0 0 1 WA2WNI-7
```

Note that you can manually remove other nodes, even non locked ones, at least temporarily, by using this command. The values you use for quality and obsolescence are not important. Port and neighbor must be entered exactly as stored in the nodes list.

The difference between the lock and unlock commands is the minus sign. Setting the obsolescence to zero permanently locks the destination node into your node routing table. Even if the locked node fails, it will still be listed in the node routing table. A failed node entered as a locked *route* on the other hand, will not be

listed in the node routing table if a corresponding locked nodes command has not been used.

Note that if 3 neighbors report a higher quality than your locked quality in a locked node, that your locked entry will be shoved off the nodes list and will not be remembered.

The locked nodes command to use if a node should NOT have an alias is:

```
N <nodecall>+ * <quality> <obsolescence> <port>
<neighborcall> (digicall1, digicall2)
```

Usage example: **N AK7Z-1 + * 143 0 0 AK7Z-1**

ROUTES

This command allows the sysop to modify the neighbor table.

This command allows routes to specific neighbor nodes to be locked in or changed. There may be times a node-op will want to modify the path quality value of a route to a given node.

To permanently add a route to the neighbor route table the command is:

```
R <port> <neighborcall> (digicall1, digicall2) +
<pathquality>
```

port is either 0, 1, 2 or 3.
0 means *route over radio port*
1 means *route over RS-232 port*
2 is a special function which is used to block node information for a certain callsign + ssid.
3 is a special function which is used to restrict access by a certain callsign.
nodecall is the callsign of the neighbor node whose route you are adding
digicall1 and 2 are optional digipeaters in the path you are adding
pathquality is the value that will be used in the node routing broadcast interpretation. See *Nodes Broadcasts* in the *Theory of Operation* section.

To unlock the routes or to change the quality value for an non-locked route use:

```
R <port> <neighborcall> (digicall1, digicall2) -
<pathquality>
```

When unlocking a route all of the data in the command must exactly match the locked data. To change the quality value for a non-locked route simply specify the new PATHQUALITY.

An optional use of up to two digi's can be specified. An example:

```
R 0 N2CGY-3 + 143 WA2JWJ-1 W5ODA N5AA
```

The result of this operation might look like:

```
MAL:W2RRY-1 Routes}
0 DKC:W5YI 50 9
0 CLOUD:W2VXY 192 43
0 WMASS:N2CGY-3 via WA2JWJ-1, W5ODA, N5AA 143 1!
```

Here we see the exclamation mark which indicates a locked entry.

The most common example of locked routes is in a backbone link which is supposed to be protected and dual ended. You may lock in the neighbor route and set the radio channel 0 path quality (Parm 3) to zero. This protects against unauthorized backbone use or misrouting caused by propagation or DX. The wanted routes would then be locked in at quality 203. This means that all nodes sourced from the neighbor will have routing qualities based on the 203. See *Theory of Operation* for more information on quality calculations.

```
0 DKC:W5YI 50 9
```

In the routes list the second value after the neighbor callsign (in this case =9) is the number of nodes sourced from the listed route. If a route is locked this value may be 0, indicating that no nodes are sourced from the neighbor.

Changing the value of an established (but not locked) route may also be done with the routes minus command. Note that attempts to remove a route which is sourcing nodes will not be effective. The best you can achieve is to set the route quality to 0. If a node is locked using a route you want to remove you must first unlock the node. If a node is locked to a neighbor for which there is no route, a route will be created automatically at the quality with which the locked node is set.

RESET

This command causes the routes table and nodes table to be cleared, along with the info message. Any currently connected users are lost. This command should be run after changing out the firmware EPROM or if any problems develop that can't be solved by other means and must be attributed to the TNC's digital section.

Since this command must be done over the air and since the RESET command causes an immediate CPU reboot the node will not disconnect you, or acknowledge the command. You will find yourself hung and must manually disconnect. Eventually the origin node you used will time out or detect a failure and disconnect you. Remember to put back the info text and any custom parms after you reset a node.

Node Parameters

The values given in [brackets] are current default for NEDA NetworkU=USER B=BKBN type ports.

1. Minimum path quality for automatic updates.

This parameter sets the minimum value for quality for a node to be saved into the routes table. When a node hears a nodes broadcast from a neighbor node it processes that broadcast in terms of the quality value associated with that neighbor node. Any nodes learned about whose resultant quality is less than parameter number 1 are ignored. If the path quality to all backbone nodes is the same, regardless of path type or port number the length of the network can be predicted based on the path quality and parameter number 1. NEDA recommends 50 for this value and 230 for all path qualities. This limits the duration in terms of number of nodes to 8, multiport, dedicated link supported, nodes.

(Range: 0 - 255) U - [50] B - [50]

2. Path quality assigned to radio channel 0 (HDLC port).

In a system where time to live (Parameter 18) is used to determine the number of backbone hops away from which this node will be available this parameter, HDLC default quality, should be consistent on all backbone ports. The NEDA Network consistently uses 203. For *user/gateway* ports a value of 50 is used so that nodes appearing from outside the network will show on the local user port. On LAN ports this parameter is set to 0 so that miscreant nodes do not show up.

Note: Changing this value will not change existing route qualities. This will only affect new routes. To change existing routes you can:

manually use the R command in sysop mode;
decrease broadcast interval for a short time so that the current routes expire;
disconnect the node from the radio so that current routes expire

(Range: 0 - 255) U - [0] B - [203]

3. Path quality assigned to RS-232 channel 1.

This value is set to 203 on all TNCs so that time to live may be used as noted above. Do not set this parameter to 255 as in a 3 or more port node this can cause feedback loops where bad node information may linger for extended lengths of time. See NEDA Annual Membership Package for more information on path quality and time to live parameters.

Note: Changing this value will not change existing route qualities. This will only affect new routes. To change existing routes you can:

- *manually use the R command in sysop mode;*
- *decrease broadcast interval for a short time so that the current routes expire;*
- *disconnect the node from the radio so that current routes expire*

(Range: 0 - 255) U - [203] B - [203]

4. Obsolescence counter initialization value.

Each time a neighbor makes a nodes broadcast this node stores the information that broadcast contained. Along with the node callsigns, names, neighbor node and quality is stored a obsolescence value. The obsolescence value is initialized to the value specified in this parameter.

Each time this node makes a nodes broadcast the obsolescence values for all stored node listings are decremented by 1. Each node listing whose obsolescence value is decremented from 1 to 0 is removed from the routing table.

(Range: 0 - 255) U - [3] B - [3]

5. Obsolescence counter, minimum for broadcast.

This sets the limit on the minimum obsolescence value associated with each node for it to be included in the nodes broadcast. The node doing the broadcasting is always included in the broadcast. This value used in concert with *Obsolescence counter initialization value* can be used to force a node to only broadcast itself by simply making this parm bigger than the initialization value. This is a desirous effect for ports facing nodes which don't participate in the *dedicated link* backbone system.

(Range: 1 - 255) U - [4] B - [1]

6. Broadcast timer interval in seconds.

A TheNET node TNC has an internal broadcast interval timer. This value sets that timer. When the timer runs out the node decrements the obsolescence counts for all of the nodes in it's nodes table and does a node broadcast. The nodes broadcast is a formatted list of all nodes in the routing table.

Whatever value is set, it should be the same as that of the neighbor nodes. If this node broadcasts more frequently than the neighbor node it will forget about listings that the neighbors tell it about. This is because the obsolescence counts may be decremented past 1 before the neighbors rebroadcast. The opposite is true if this node doesn't broadcast often enough.

(Range: 0 - 65535) U - [1800] B - [1800]

7. Link time-out (FRACK).

This sets the time delay after a message is sent to a user or neighbor node before the node will attempt a retry. For double ended hidden transmitter free backbones this should be set to a minimum value, 1. For *user* ports, setting this value higher gives priority to those users that are most consistent into the node. Higher values (8 - 12) should be used on user ports if users are likely to digipeat to the node.

(Range: 1 - 15) U - [3 to 6] B - [1]

8. Link layer MAXframe.

This parameter is the same as the MAXframe command available in most user TNCs.

This sets the number of packets, of those that are available in memory to send to a user or adjacent node, that will be sent in one transmission. In all cases this should be set to 1. Setting this to a higher value will intermittently allow some users a higher percentage of network and user port bandwidth without substantially altering total network efficiency..

(Range: 1 - 7) U - [1] B - [1]

9. Link Layer Maximum retries.

(Range: 0 - 127) U - [6 to 10] B - [8 to 14]

Depends on channel loading / sharing characteristics

10. Digipeating.

If this function is enabled(1), it allows users to subvert the normal network flow by assigning priority to the digipeating station. The default of disabled(0) is recommended.

(Range: 0 - 1) U - [0] B - [0]

11. Validate callsigns.

This option allows the sysop to enable or disable callsign validation. This affects the C command in the nodes command processor. This also affects the valid callsigns that may be used to connect *to* a node. If this feature is enabled a user would not be able to connect to the node with the callsign of N0CALL.

If validate callsigns is turned on the C command in the node will only accept valid callsigns and existing node names. This feature should be enabled in all nodes as this eliminates the delay use user might have to wait to find out that he/she specified a node name incorrectly.

(Range: 0 - 1) U - [1] B - [1]

12. Host Mode connects.

ON (1), OFF (0).

When a NodeOp has a terminal connected via the RS-232 Host Interface, ON (1) will allow users to connect to him IF he is not actively connected to the node at the time. Off (0) prevents users from connecting.

(Range: 0 - 1) U - [0] B - [0]

13. Node radio TXD.

TXDelay in a TheNET node is the same as TXDelay in a TNC2. This adjusts the period of time between keying the transmitter and when it actually starts sending data. If this value is too short the receiving station will not hear the start of the packet and a failure will result. If this value is too long then data throughput will be less than optimal. TXDelay is adjustable in 10 MS increments.

(Range: 0 - 255) U - [35] B - [5 to 35]

User ports should not be shorter than 35 or you will exclude stations with slower switching radios from using your user port. Backbone ports should be optimized to find the absolute lowest value that will work reliably with all other radios on its backbone link, then bump the number up a few notches so switching delay drift doesn't interfere with reliable Tx / Rx switching.

14. Broadcast via port.

Nodes broadcasts occur once each halfhour (depending on nodes broadcast interval parameter). This parameter allows the sysop to disable nodes broadcasting on one or both ports. The reasons that this might be done is to • discourage node operation on the radio port and reduce clutter on the frequency or • hide the node or • in concert with locking the node in at another location this can be used to create a gateway or dedicated use link.

0 = Broadcasts disabled on all ports.

1 = Broadcasts enabled on port 0 (radio) only.

2 = Broadcasts enabled on port 1 (RS-232) only.

3 = Broadcasts enabled on both ports 0 and 1

(Range: 0 - 3) U - [2,3] B - [3]

15. Hidden Node Propagation.

This causes # nodes to either be propagated or not. # nodes are usually used for backbone links. In the NEDA network there are about 60 user ports and about 100 backbone ports. As the TheNET node can only afford to have about 90 nodes in it's node routing table it would be impossible for a single TheNET node to have the entire network in its table at the same time if # nodes were to be listed. This command is normally left off in the NEDA network. Also keeping this parameter turned off reduces the length of a nodes broadcast if there are any # nodes in the local system.

(Range 0 or 1) U - [0] B - [0]

16. Connect Command Enable.

If set to 0 connect commands typed after connecting to a node are quietly ignored. This prevents stations from doing manual L2 connects from a backbone node.

(Range 0 or 1) U - [1] B - [0]

> *Note: Parameters 17 through 34 are only available before burning the EPROM. Only parameters 1 through 16 are available over the air.*

17. Maximum number of nodes in NODES list.

Both hidden and non-hidden, in the node routing table. If this number is set too low, say to 1, you will limit the number of neighbor routes that show up in your ROUTES table. In other words, your node will not recognize more than one network node.

(Range: 1 - 400) U - [100] B - [100]

18. Time to live.

This is the number of node hops that a message from this node can go before it is killed. Each message transmitted through the network by a node has an associated time to live. Each time the message is received and retransmitted by any TheNET node the time to live for that packet is message is decremented. If the time to live reaches 0 the message is thrown out. This parm sets the time to live start value for each message originated by this node.

(Range: 0 - 255) U - [7] B - [2]

19. Transport layer time-out.

Sets the number of seconds that your local user port will wait before retrying a packet across the network. In this time the destination user port must acknowledge the packet and that acknowledgment must make it back to the origin user port. If the packet is retried there will be a second redundant copy of the message heading across the network even if the first copy successfully arrived. This value must be set to the maximum amount of time that it will take for a packet to travel the number of hops as set by parm 18 and parms 1, 2 and 3 and for the acknowledgment to return to the node of origin.

(Range: 5 - 600) U - [200] B - [200]

This gives a 3.33 minute time-out

20. Maximum transport layer tries.

Sets the number of copies of a given packet that the origin user port will send into the network to the destination user port, timed by parm 19, before declaring the path disconnected. Hopefully the TheNET software in conjunction with nodes routing information will be able to try alternate paths to achieve a response from the destination node.

(Range: 2 - 127) U - [2] B - [2]

21. Transport layer acknowledge time.

This is the amount of time a port waits before acknowledging a transport layer packet that was received. Faster is better here unless the port will be under continuous heavy loading from the same destination node.

(Range: 1 - 60) U - [1] B - [1]

22. Transport layer busy delay.

In the event that a transport layer circuit cannot handle more data (parm 24) a busy flag is generated. This parameter is the number of seconds that the origin node waits before retrying a message that was lost due to the busy condition. When the busy node clears it also generates a packet back to the origin node announcing that it is clear.

(Range: 1 - 1000) U - [180] B - [180]

This equals 3 minutes

23. Transport layer window size.

This is the number of unacknowledged packets that can be outstanding for a given circuit (each user connect).

(Range: 1 - 127) U - [2] B - [2]

24. Congestion control threshold.

This is the number of packets that can be buffered in a user port for a given circuit.

(Range: 1 - 127) U - [4] B - [4]

25. No-activity timer.

This is how long a user may stay connected with no traffic flowing between the node and his station. On version 2.05 and later also sets the life of the node STA light following the last user disconnect.

(Range: 0 - 65535) U - [7200] B - [300]

The higher value on a user port here allows users to hang out on CROWD ports and special servers i.e.: DxClusters for extended periods without being tossed off for lack of activity.

26. P-persistence.

This figure determines the aggressiveness of the TNC's transmit function. A high value of P-persistence will cause the TNC to be very aggressive. If there are more than 2 TNCs waiting to transmit on a single channel and their P-persistence is set incorrectly the data throughput will suffer. A formula used to calculate ideal P-persistence is Pp = (256/N-1)-1 where N is the number of TNCs on the channel that could have data to go out at the same time. So, if the channel only has two TNCs the P-persist could be set to 255. If the number of TNCs is 3 then P-persist could be set to 127.

(Range: 0 - 255) U - [32 to 255] B - [255]

User ports depend on type of port and if it is shared by other nodes within direct RF connect range. Backbone port value assumes dedicated end to end HTS free link.

27. Slot time.

This value should equal the TXDelay for the node plus the worst response delay for other stations on the frequency. For dedicated point to point links (2 radios on a frequency) this value is unimportant as P-persistence when set to 255 overrides the value of slot time. As with parm 26 this value depends on the type of application. Parameters 26 and 27 work together to set up a random delay determining when the node will key up following a DCD decision that the channel is clear. This is an anti-collision technique. When the node is ready to transmit, a number between 0 and 255 is internally generated. If the random number is equal or less than the value set by Parameter 26, the node keys immediately upon sensing a clear channel. If the internally generated number is greater than the value of Parameter 26, the node waits for a period of time equal to the slot time and then internally generates a new number, etc. A value of 63(+1) is 25% of 255(+1) and thus sets the percentage of time the node will immediately keyup. Protected trunking nodes (those with only one transmitter on their receive frequency) would have faster throughput if there were no node keyup delay. Setting parameter 26 to a value of 255 will accomplish this.

(Range 0 - 127) U or B - [TXDelay of radio]

28. Link Layer time-out (Resp Time).

10s of milliseconds between receiving a packet from a neighbor node or user before the node will acknowledge a packet. This is actually the *response* time in Ms. Setting this value too low on a user port will prevent some users from being able to access the port as older radios and some newer rigs with very slow locking synthesizers will not recover from transmit fast enough.

(Range: 0 - 6000) U - [50 to 100] B - [15 to 35]

29. Link time-out timer (CHECK).

This parameter sets an idle link timer. If a link is inactive for this amount of time a check packet is sent to make sure the other end is still there.

(Range: 0 - 65535) U - [65000] B - [65000]

30. Station ID beacons.

Either ENABLED (2), CONDITIONAL (1), or DISABLED (0). This directs the node to ID either every 10 minutes, only after activity, or only imbedded in AX.25 packets. If the node is addressed using the mnemonic by users it will not be properly identified with the AX.25 packets and must be set in either the ENABLED or CONDITIONAL for user ports. For backbones each node is only addressed by callsign so ID beacons may be disabled.

(Range: 0 - 2) U - [1] B - [0]

31. CQ broadcasts.

ENABLED (1), DISABLED (0). Disabling this feature means the unproto CQ user text will not be broadcast by the node. The CQing user will still be able to be seen by someone doing a USERS command during the time the CQ is active.

(Range: 0 - 1) U - [1] B - [0]

32. Heard list length.

Sets the maximum limit on the number of stations that can be listed in the Heard table.

(Range: 1 - 20) U - [20] B - [20]

33. Full Duplex

ON (1) or OFF (0)

This option is only turned on if a full duplex radio set is employed for a backbone.

(Range: 0 - 1) U - [0] B - [0]

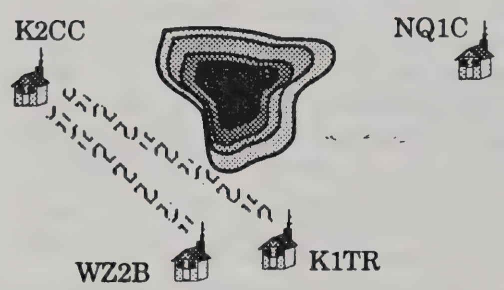
Networking Around HTS

AX.25 is a Carrier Sense Multiple Access system. That's what CSMA stands for. CSMA means that each station depends on it's own receiver to determine when it's OK to go into transmit mode. In many commercial packet systems which use CSMA type protocols it is given that all of the transceivers can hear each other.

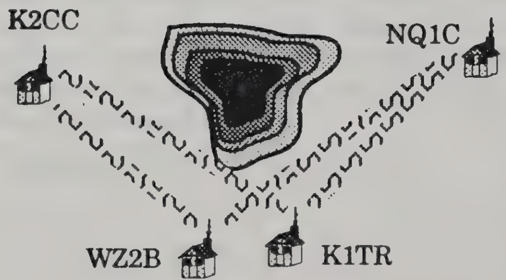
In amateur radio packet that is almost never the case. What that means is that sometimes a station can be talking and another will just come onto the frequency and start talking. It sort of sounds like twenty meters on a winter Sunday evening. Usually on twenty meters when that happens, an operator who can hear both stations that were transmitting will say something like "The frequency is already in use".

In AX.25 packet what does happen is that the two stations who were talking don't get any answer so they try again. Often times the timing can work out so that both stations will once again transmit at the same time (collide) and will waste even more time. What is worse is that in many areas on two meters there will be more than two stations trying to transmit at a time. There may be dozens. This means that the number of bad transmissions per successful exchange can be very large. Each time there is a bad transmission the stations have to wait a certain amount of time before retrying also. This wastes time.

Most amateur radio packet on two meters is done at 1200 bauds. This means about 150 characters per second. If there are only two stations in the local area and they can only hear each other, and they have reasonably fast radios the number of bytes that they can transfer per second is around 80. In a situation with two stations who can't hear each other, trying to pass data to another two stations who hear both equally well, the rate will probably be closer to 5 bytes per second per station. 80 bytes per second is pretty fast for a person to read. 5 is very slow. If the two stations could hear each other the rate might be up to 36 bytes per second for two stations. That's assuming that they are not both 'greedy'. If they are both 'greedy' it's possible that no data would be passed at all! The process by which AX.25 (CSMA) stations jam each other because they can't hear each other is called Hidden Transmitter Syndrome or HTS.



In this illustration we have K2CC talking to WZ2B. K1TR is listening but is not involved in the conversation. The throughput between K2CC and WZ2B is about 80 characters per second.



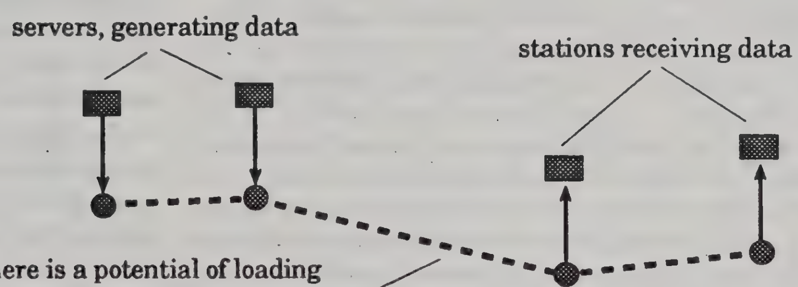
In this illustration we have K2CC talking to WZ2B. NQ1C is talking to K1TR. Because K2CC and NQ1C can't hear each other they frequently go into transmit mode at the same time. K1TR and WZ2B both get garbage. Throughput is drastically reduced. NQ1C and K2CC are called Hidden Transmitters because they can't hear all of the transmitters that other stations they talk to can.

Network Implications of HTS

100% of theoretical throughput could only be arranged if there was some overriding control that made sure that there were never any collisions and that the transmitters came on just at the right times. This is not possible if the only means of synchronization is via the CSMA channel and random timers. If all things are arranged as best they can be the best performance that can be achieved with a hidden transmitter environment is 20% of theoretical throughput. Two notes. 1. Only if all participants on the channel are cooperating will this occur and; 2. Only if the participants don't try to push the throughput on the channel. If the throughput exceeds a threshold (determined by the aggressiveness of the users) then collisions and retries on the channel will force the throughput to near 0%. In order to prevent the throughput collapse, means of backoff must be implemented. Because backoff is not a feature incorporated into most AX.25 based devices the only way to prevent throughput collapse is to avoid having hidden transmitters that source data.

Workable Methods of Avoiding HTS

For backbones the best method of avoiding HTS is to have no hidden transmitters. If there will ever be a situation where two stations were sourcing a data into a particular link, even from further down the network, then that link should be a double ended point to point link.



Because there is a potential of loading from more than one source on this backbone it should be a point to point link, i.e. only two radios on the frequency that can hear each other.

For network access points the same rules apply. If there will be more than one source of data cranking out at full speed then there should be dedicated links to each source of data. It is important to note the difference between data generators and data receivers. A data receiver isn't a potential part of HTS. You can create a situation where there will be a dozen data receivers, hungrily accepting data and only generating the briefest of acknowledgment transmissions, on the same frequency. If, however, there is even one hidden source of data on the frequency, the data receivers will not be able to get even the short acknowledgment through.

Most packet users spend most of their time receiving data. This is great because it fits the model of *data receivers only* just fine. Even when the average packet user is in data generate mode the frequency of packet data transmission is rather low. Usually the user is hand typing messages. This fits well within our 20% or less loading theory.

AX.25 in the Network

So what we have decided is that there are a few different ways that TNCs and CSMA are used in the network. There are point to point backbones, low usage hidden transmitter free backbones, hidden transmitter free input ports for data generators, and user channels for data receivers only.

Node Sites and Hardware

The best place to put a node is where it is most convenient. If it's easy to build a node it is more likely to get built. If a node needs to be constructed to serve a particular link and only one site will do, then built it for that site. There is a lot to be said, however for having nodes situated where they can be observed, serviced, and played with.

Higher is not better

There may be several reasons for putting up a node. One of those reasons is to allow a group of packeteers to access services or other stations that are available through a network of nodes. Perhaps you already have a network to link into, or perhaps you have hopes of building one. At any rate user access to the network is important as the users are the best candidates for creating new services and new network installations in the future.

The best user access to a network would be where each user has a dedicated point to point link from the network node to the user station. This is extravagant to say the least. A good compromise would be a low coverage user port that serves only a small number of users. The limitations and design goals for a user port are that there should be no more than twenty stations on the air simultaneously accessing the user port and none of the stations should be major sources of data. Very infrequently should the users of your user port generate data faster than typing speed. Most users spend most of their on air type receiving data from BBSs, DxClusters, CROWD nodes or databases so this isn't much of a limitation.

Because your user port can't be allowed to hear any other node sites (or servers) your coverage is going to have to be strictly controlled. If your node site is on top of a high mountain or tower this may be difficult. Use of directional arrays or low gain antennas may be required. Perhaps an attenuator or tight squelch could solve the problem. Keep in mind that your node's user port shouldn't be allowed to be heard by the other nodes on the frequency either. Sometimes a node site that is designed to serve a distant city or long river valley can take advantage of tight patterned yagis. Keep an open mind and don't use high power. Remember that amateur radio spectrum is a limited resource. Use it wisely.

Node sites in homes have particular advantages. Whenever a ham is involved certain characteristics may be assumed. One is that if three radios is fun there will bound to be five or six in the near future. Putting a three port node in a ham house situation is a good way to make sure that more expansion occurs. These things are darn fun to watch. It is also particularly easy to add local computer access to the node with minimum expense. This means that a server can be added to the network very easily.

Nodes in commercial sites have advantages as well. One of these is that backbone paths can usually be quite long. Often commercial sites offer fairly high towers so separation between antennas on the same band may be achieved. It is quite possible to run as many as a half dozen backbone links in the UHF ham band at a single site. The way this is done is by running the links in half duplex mode. The receive and transmit frequencies may be split by as much as fifteen or twenty megahertz. Then the links can be set up so that all of the receivers are in the high end of the band and all of the transmitters are in the low end of the band. So long as the antennas are reasonably separated vertically this should be very easy. Because your radio's transmitters are about twenty megahertz away from the commercial band this may be easily approved by the site managers. This is one of the more wild ideas for node to node linking. Using 25 watt commercial or amateur mobile radios on simplex you should be able to get two or three UHF links at the same commercial site.

One problem associated with commercial sites in some metropolitan areas is that the coverage for the user port may be higher than desired. The easy solution to this problem is to not have a user port at the high node side. Perhaps one of the pre-existing servers would house a user port. Perhaps you can set up several by using dedicated links to each of the local servers in the metropolitan area and maybe adding a couple of node sites just for the sake of having low coverage user ports. If your commercial site has good enough coverage of the city your cellular user port/nodes can be made using low power handy talkies. One watt commercial UHF handy talkies can be readily had for less than $100. Used two meter ham gear is pretty cheep. A simple UHF antenna, a two meter vertical, two feedlines, two TNCs and a power supply is all that is required to make a cellular user port. Now that you've got all of these simple repeater sites located in peoples homes, how long will it be before some more backbones start showing up into these sites. Your system will expand quickly as the ham radio public realizes how much fun it is to play with a real packet network.

Radios for Nodes

Radios selected for node use should be capable of heavy duty use. The Tx/Rx switch circuitry should be able to handle virtually millions of operations without failure. This means PIN diode Tx/Rx switching as a first choice followed by high quality reed relay switching. Receiver front-end filtering should be quite sharp if your node is to coexist with other radio services. In that case consider using one or two tuned cavities to cut down on

front end overload and desensitization. If your radio is operating on a simplex frequency, the cavities will also aid in reducing the effects of "white noise" being generated by the transmitter. At congested sites, a circulator may be required.

Some amateur class VHF radio's employing PLL frequency synthesizer technology should be avoided. Two reasons: PLL settling time between transmit and receive is too slow for optimum packet throughput. Consider the following table.

This is a table of maximum throughput in bytes per second assuming 230 bytes of data per 256 byte transmission with a 16 byte acknowledgment, for each popular data rate and with different TXDelay settings which would be the same on both ends of a data link:

baud rate	byte time	Throughput given TXD of 0ms	40ms	250ms	350ms	500ms
1200	6.67ms	127	121	99	91	81
2400	3.33ms	254	233	163	143	121
4800	1.67ms	506	431	241	199	158
9600	.833ms	1015	750	316	248	187
56K	.104ms	8131	2124	435	316	223

Note that the actual TXDelay setting in the TNC Parms is in tens of milliseconds. Therefore the 500ms values in this chart would be achieved by setting the TXDelay to 50; 40ms values would be TXDelay of 4.

The length of a single byte BYTELENGTH = 8/baudrate
The length of one byte of data, including inefficiencies is
LOADEDBYTE = [(TXD x 2transmissions) + (BYTELENGTH x 272bytes)] / 230

Throughput per second = 1 / LOADEDBYTE

This means that the speed of the radio's transmit to receive and receive to transmit switching is vitally important. Also, the transmitter may be keyed before stabilizing on frequency. This latter situation could cause interference to other receivers on different frequencies. This may be a serious concern if you choose a commercial radio environment for your node. If your candidate radio uses PLLs, solicit the manufacturers advice on suitability for packet node use.

In general, retired commercial service FM radios, such as the Motorola MICOR and GE MASTR II, or later, make excellent node radio choices. The commercial radios are designed to operate in moderate to high intensity RF environments, are physically rugged, and fairly reasonably priced on the used market. These radios typically come in a variety of power levels up to 110 watts (suitable for long haul dedicated UHF/6m backbone links; User ports should generally run less than 25 watts ERP.)

If this information is daunting to you then please just keep it in the back of your mind. If you are running your node out of a non-commercial radio site, like your home, then you can worry about this after you have your multiport node up and running. Using ham radio HTs and mobile rigs you can get things going and then swap out critical components later. The most important thing here is that you get your multiport node up and running with dedicated point to point backbones. Then you can worry about radio and baud rate improvements.

TNCs

MFJ 1270B and PacComm Tiny 2 are the current models of the chosen TheNET TNC. Neither TNC needs modifications to work with TheNET. However, there is a bug with the MFJ1270B in that some models are shipped with the RS-232's control lines messed up. The general fix for this is to jumper pins 20 and 4 on the RS-232 connector for that model. If you use a Tiny 2 you can operate at 19.2Kbaud with the HexiPus™. Also the HexiPus™ pinout is identical to the Tiny 2 so you can use a straight through cable.

Finally

Note: Don't compromise on anything. Be as high class in your system design as you can and still get results. This way your system will expand gracefully. If you compromise on your backbones and don't use point to point links you'll hurt your network very badly later.

TheNET Node Mnemonics

TheNET nodes incorporate a translation table that allows each TheNET TNC to be referred to by a six character mnemonic. This mnemonic is called the *node name*. Node names exist for the convenience of the human users. When ever a connect is made from a node to another node the node callsign is used, even if the user specifies the node name. So, whenever traffic is monitor between two nodes on a backbone the nodes refer to each other by callsign.

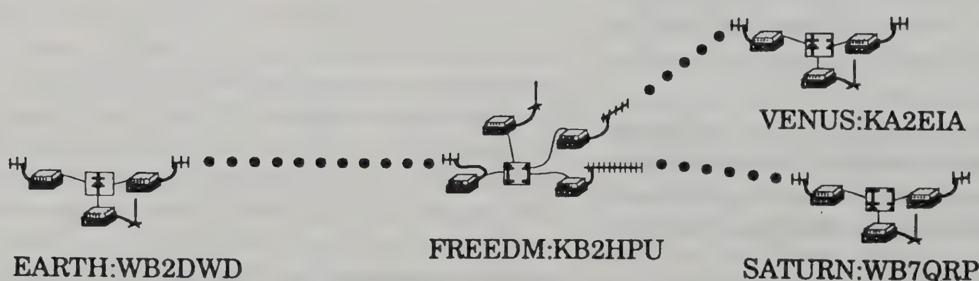
TheNET nodes run in TNC 2s. Thus there is a limit to the amount of memory available both for program and for data. The number of nodes that can be listed in memory is limited. Normally this is limited to 100 nodes. A feature of TheNET is that nodes that are used for backbone links can be made to not propagate automatically. That way the only nodes stored in each TheNET TNC's routing table are the user connect ports or node ports that must be visible for other reasons. The way that backbone nodes are set to not propagate is to name the node with a "#" character as the first letter in the node's name. A second feature of TheNET is that when a user requests a nodes list with the *NODES* command the node does not list the # nodes. Thus they are called hidden nodes.

Backbone Ports

Even though backbone port nodes won't be visible with the NODES command they can still be seen by using the NODES * command. Also when a ROUTES command is done the backbone port nodes will show up. Generally the backbone nodes are given full six character long identifiers that have the first character as a #. If the backbones are implemented on dedicated point to point links then they will only be seen by one other station. Because they have info texts that define where and what they are is is usually pointless to name the backbone nodes with the same name as the user port, i.e.: CANTON, #CANTN, #CANT2. One of several other node name solutions should be used.

One choice for naming # nodes has been to use the adjacent node's name as the node name for the backbone port.

Thus in a network that looks like:



if we looked at the Routes command response from FREEDM we might see:

```
FREEDM:KB2HPU-1} Routes:
> 1 #EARTH:KB2HPU-12 203 38
  1 #VENUS:KB2HPU-10 203 37
> 1 #SATRN:KB2HPU-11 203 29
```

This makes it pretty clear where one would have to connect to look at the actual backbone TNC to go to, say, VENUS. The only problem with this is that if you went to #VENUS:KB2HPU-10 and did a routes command you'd see:

```
#VENUS:KB2HPU-10} Routes:
> 1 #EARTH:KB2HPU-12 203 38
  1 FREEDM:KB2HPU-1  203 37
> 1 #SATRN:KB2HPU-11 203 29
> 0 #FREDM:KA2EIA-13 203 28 !
```

This is somewhat confusing.

Some clubs show their unity by naming all of the # nodes the same. Thus: #NEDA, #CNYPA ,#NEPRA might be seen on dozens of nodes. This does not cause a problem as all node to node routing is done by callsign. When a station connects to a node in the NEDA network, for instance, and does a ROUTES command, he might see:

```
CANTON:WA2MZF-5} Routes:
  1 #NEDA:WA2MZF-12 203 16
  1 #NEDA:WA2MZF-10 203 3
> 1 #NEDA:WA2MZF-11 203 12
  0 HULL:VE2RBH 50 1
```

If the user knows about NEDA, he will recognize this immediately. This does not convey as much information as the previous method. The user probably doesn't care though and this method, when used over a large network is certainly impressive.

Another method is to use a site designator for three characters, followed by the compas heading, such as #ALBAW, #ALBAE, #ALBAS as backbone nodes for the ALBANY node.

User Ports

User ports should be labeled with a town or mountain, or in bigger cities, the neighborhood.. Identifying a user port by club, region or airport is not as good.

Clubs are generally not known outside the area of it's

influence and if it's area of influence is large then the location of the 'club' node is not obvious.

Naming a node by region, like WMA or SNY or COHIO is a problem because if the node really does cover an area that large then it is probably useless due to hidden transmitter problems. Also the very existence of a node with such an all encompassing sounding name is that other people who might put up a node in the same region might feel that they are stepping on toes or that redundancy isn't desired.

Airport identifiers may be sufficient for naming nodes as far as locating them on a city by city basis but there will no doubt be cases where the airport isn't near the city that it is named after, there are many more than one node per airport, or there is no airport nearby. Also many hams would have a hard time figuring out where a node is by it's airport identifier.

By all means name a node by it's city rather than by some cute code that's useless to all but the naming party.

The user port need not have the frequency in it's title. A node named ALB144 is not as obvious as one named ALBANY or ALBNY1.

Here is a system for compressing a long city name into six characters which could also be used for compressing into five characters. This method was submitted by VE1YZ. VE1YZ gives credit to Boeing Corp for documenting this standard for the aircraft industry.

Names with the desired number of letters or less are left alone.

Names with more than the desired number of characters are abbreviated using the following rules sequentially until the desired number of letters remain.

- Delete double letters.
- Keep the first letter, first vowel and last letter. Delete other vowels starting from right to left.
- Keep the last letter, then delete consonants from right to left.

Fixes with multi-word name

- Use the first letter of the first word and abbreviate the last word using the above rules sequentially until only the desired number of characters remains.

Examples for six character node names:

```
Albany -> ALBANY
San Francisco -> SFRNSO
Seattle -> SEATLE
Manchester -> MNCHSR
Syracuse -> SYRACS
```

Specialized Ports

For dedicated links and server ports, it is a good idea to pick a label that fits either the application, or to use a abbreviated version of the site's main user port with a number after it. *Make sure the info text clearly spells out what the port is for.*

Examples of specialized ports in our network are: DXCLUS at UTICA node which is the DxCluster uplink, and DXKNOX at KNOX which is near Albany NY and serves the YCCC/AARA DxCluster run by K2TR. Give consideration to where your label will show on the list. This was one of the reasons that BBSxxx was chosen for G8BPQ and MSYS BBS ops, so that the BBS's would all be listed in the same part of the nodes table!

Common Problems

Read this before you launch the missiles.

• • •The TNC seems to be operating at an extremely low baud rate or it takes forever to transmit.

The problem may be in the baud rate generator or the baud rate may not be selected for one of the ports. Check the jumpers if it's a PacComm. If it's an MFJ it may be a broken switch or the switch may be set to an improper value.

• • •If the node is doing funny things like:

- only working over the radio or only working over RS-232
- not responding at all
- using the wrong callsign or node name
- transmitting all the time
- not handling the LEDs right
- disconnecting everybody once in a while
- never nodes broadcasting or ignoring incoming broadcasts entirely

Try sysopping the node and using RESET or if you have local access you can turn off the TNC, remove the battery jumper for more than 30 seconds, replace the jumper and then turn the TNC back on.

• • •If your new node stack doesn't allow you to connect across the matrix, make sure that you have waited long enough for nodes broadcasts to work. The nodes broadcasts over the matrix happen at the same time they do over the radio. Just because the TNCs are stacked they don't necessarily know about each other. It is possible using the Sysop:NODE command to lock in each port in your node stack to each other port so they begin communications immediately. This is a good way to get used to doing sysop commands.

• • •You've changed the parameters so that the default route quality is 200 but the routes are still at the old value.

The route qualities are set by the parms when the route first comes into existence. If you want the route quality to change you must either change them manually using the sysop route command or by making the routes go away and then come back. This can be done by increasing the nodes broadcast rate (parameter 6) temporarily or by disconnecting the radio or matrix for several nodes broadcast intervals.

When a neighbor node is first heard the route quality is set based on the parameters. Simply changing the parameters does not change the route quality. However, if you change a route quality to a neighbor node, the nodes sourced from that neighbor node will change in value as soon as the next nodes broadcast is heard from that neighbor.

• • •If you have any other problems or figure out a problem for yourselves make sure you send a message about this to NEDA@K1MEA attn.: tech documentation.

Glossary of Packet Networking Terms

AFSK
Audio Frequency Shift Keying. This is a mechanism for sending digital information over a radio. A signal 0 is sent using one tone while the signal 1 is sent using a different tone. This is the mechanism used by telephone modems and packet radio modems.

Autoforward
Many of the bulletin board and mail server programs (BBS) are capable of passing messages to each other. The process of a bulletin board recognizing that it has mail to go to another bulletin board, connecting to another board and then sending the traffic is called Autoforwarding. (See also Forward File) This allows packet users to send mail in a non real time fashion anywhere on the planet where compatible BBSs exist.

Autorouting
This is a process by which a network node can pass traffic to another node via one or more intermediate nodes.

AX.25
This is the designation for the protocol used by TNCs to talk to one another.

Backbone
A backbone is a system of links where nodes may communicate without interfering with or being interfered with by local access, and where data may be passed in a fashion and with hardware that is optimized for passing data, rather than optimized for inexpensive user stations.

Example:

- Most user stations operate at 1200 baud on two meters. A backbone would be more efficient and less susceptible to interference if it were on UHF or 220. Also a backbone might be optimized by taking advantage of the knowledge of all radios on each link. Such optimization might included setting acknowledge delay (RESPTIME), Transmit lead time (TXDelay) or persistence (PPersist) to values that work best for the radios on the backbone frequency. Such settings might be impossible if any average user stations were to be able to access the link radios. In addition baud rates might be increased if only a few radios/TNCs need be affected.

Backoff
When a packet is sent and not responded to, the sending station will wait a specified 'backoff' before retrying. In simpler systems this is called "FRACK" or FRame ACKnowledge delay. In more complex systems, like TCP/IP, the backoff time can be calculated based on previous performance of the link. One such backoff procedure is called *exponential backoff*. In this system the amount of time delayed between resending a missed packet increases by a stable factor each time the packet is tried, until some maximum backoff time is reached. (See TCP/IP, Retry)

Baud
Baud is a measure of data flow. One baud indicates one transition of data. 1200 bauds indicates 1200 transitions. In packet radio one character of text equals 8 bits of information. 8 bits of information requires 8 transitions of the modem tones. That means that there will be about 150 characters of data in one second of 1200 baud transmission except for the fact that packet transmissions include key up time and callsign information which take up many characters of time.

BBS
Bulletin Board System. This is a server which is accessed by packet stations to be a repository for messages and files. Those messages and files can be accessed by all packeteers who connect to the BBS, if desired. BBSs also have a capability called *Forwarding* which may be used to send files between BBSs. See *AutoForward*.

Breakout Node
This is a node that is capable of handling many links. In many cases packet nodes have been installed in places where many radios or backbone links are not allowed, such as on high mountains of great commercial value. A breakout node holds no special meaning except that it is a node that has proven to be very expandable and at which the node owner sees little or no limitations on reconfiguration.

Callbook Server
This is a network server whose function is to allow stations to access, in real time, callbook information. These servers are operated both stand-alone and as part of DxClusters. See the server listing in your Quarterly.

Choke/Unchoke

When a computer is unable to process data as fast as another computer is sending it the receiving computer may instruct the sending computer to stop sending the data. This condition is often referred to in the packet world as 'choke'. 'Unchoke' refers to the re-enabling of the sending computer.

Example:

- A TNC running TheNET is capable of storing a fixed number of packet frames in memory. If this number gets exceeded data might be lost. Because each TheNET node is capable of supporting many users and some other network management functions simultaneously the memory is partitioned to smaller blocks called buffers. Each user on the node is allowed four buffers (in the NEDA network). When those four buffers are used up the TheNET TNC attempts to choke the user. If the TheNET TNC is at the far end of a network circuit the choking and unchoking process takes somewhat longer. What actually happens is that when a message is passed across the network to the destination user port that destination user port may respond with a 'choke' message. The destination user port will attempt to deliver the data that is has and when it is ready will send an unchoke message back to the originating user port. Based on a time-out timer the originating user port might resend it's delayed packet even though it has not received the unchoke message.

Circuit

In a TheNET network a circuit is an assigned connection between two nodes. Each of the two nodes has information to the effect that the circuit exists. The two nodes also have a routing table from which the first element on the path to the other node may be realized but the two nodes do not know all of the intervening nodes. The circuit exists until the destination user or server or the originating user or server disconnects, or until one of the two nodes decides that data cannot be sent any more (due to L3 retry time-out or unchoke failure) or if no data is passed across the circuit during the time set by the no activity time-out. In the NEDA network the L3 retry time-out is 5 minutes times 2 retries and the no activity time-out is two hours. (See neighbor, choke/unchoke)

Collision

This is an event where a receiving station doesn't receive it's desired packet because another packet was generated by a different packet station in the same time frame as the desired packet and interference occurs. In this case the sending station will wait a backoff time and the packet is retried or a special poll packet is generated. (See backoff, poll-packet, retry)

Converse Node

See *CROWD*

CROWD

This is the name given by the NEDA founders for a piece of software written by NORD><LINK to run in a TheNET network. Most NORD><LINK documentation refers to this is a mini-conf (conference) node. The CROWD software is installed at a TNC. Access is over the network only, through the serial port in the CROWD TNC from other TNCs at the same node site. There are several CROWD nodes in the NEDA network and several more in other networks.

CSMA

Carrier Sense, Multiple Access: This is a system of packet operation that requires that all stations wait for the channel to be quite before transmitting. Most amateur radio packet uses CSMA.

Dedicated port

This is a port designated for a specific purpose with only one other station on the frequency, usually a tie-in to a server or other network hardware.

Digipeater

A TNC used for relaying messages on a single frequency. Digipeater functionality is built into all user TNCs. A digipeater is used for sending a message beyond the range of a user station. Normally if there are two stations that want to communicate beyond their own range they will use a digipeater in between. One detail with digipeaters is that they do not recognize when a message they have relayed does not get through. It is up to the sending station to retransmit. Digipeaters are inherently susceptible to hidden transmitter syndrome. NEDA recommends against digipeating in any form except in emergencies.

Diode Matrix

The TNCs running ROSE or TheNET network software can communicate to each other over their RS-232 ports. If two TNCs are used at a node site the connections are simple connector to connector wire connections. If more than two TNCs are used a diode matrix is required. (See HexiPus™)

DOVE

An OSCAR satellite (OSCAR 17) whose full name is Digital Orbiting Voice Encoder.

DxCluster

A server used by HF operators to pass information about contacts. This software, originally written by AK1A, also operates as a database of HF related information. One key feature of the DxCluster software is that DxClusters may share contact information (called Dx Spots) in real time. That is that one station connected to one DxCluster may introduce a Dx Spot report which will then be shared by all of the stations connected to all of the DxClusters which are networked together. See the server listing in your Quarterly.

Dynamic Rerouting

In a network where redundancy exists in the backbones from one city to another some types of network software allow for the network to recover automatically from a backbone hardware failure by rerouting traffic through the redundant link. This is called 'dynamic rerouting' as it can adjust dynamically to a changing network. (See ROSE, TCP/IP)

EPROM

Erasable Programmable Read Only Memory. This is an IC which is used in computers, including TNCs, to permanently hold a computer program. In PCs and Macintoshes EPROMs are used to hold the *boot program*. That's the program which is responsible for loading the operating system into the computer from a hard disk or floppy disk. In TNCs all of the program is located in one EPROM. EPROMs are erasable using Ultraviolet light for between 2 and 40 minutes. Thus EPROMs have a small lens in the middle of the top which exposes the internal electronics. During long term usage EPROMs are covered with a piece of opaque tape. EPROMs can be programmed using a peripheral to a PC called an *EPROM programmer*. They cost about $150.

ERS

Exposed Receiver Syndrome: This is a condition where a packet station, be it node or user, is unable to transmit due to the fact that it perceives the channel as being active continuously. This can be caused by Hidden Transmitter Syndrome and is often the case when a node is located on a high hill with surrounding metro areas. (See HTS, CSMA) Also see an article in this Quarterly called *Exposed Receiver Syndrome*.

False Route

In a network using TheNET software the node routing is generated automatically by the nodes themselves. If improperly managed it is quite possible for routing to be discovered and used by the nodes such that Dx paths are treated as real paths. In this case a route may be created in the routing table that depends on a 'lift' (propagation enhancement) condition. When the lift goes away the nodes will be helplessly trying the 'false route'. This condition is preventable in a TheNET system by manually controlling the route tables to specify valid routes to neighbor nodes. This situation cannot occur with ROSE or TCP/IP software as all neighbor nodes and routing information is created either manually or by software that is much more intelligent than TheNET. (See TheNET, locked route, ROSE, TCP/IP, neighbor).

Forward

See *Autoforward*.

Forward File

This is the disk file on a packet bulletin board system (PBBS) that is responsible for directing the autoforwarding operation. By making entries in this file the PBBS system may select what packet paths are used to each PBBS that is forwarded to, when each operation is performed and what traffic is sent during each piece of the forwarding operation. (See autoforward)

FRACK

FRame ACKnowledge delay: This is the time after a packet is transmitted by a TNC before the TNC decides that a frame acknowledge is not going to occur. At that point the TNC performs backoff (some TNCs + TCP/IP) and a retry. (See backoff, retry).

FTP

File Transport Protocol. This is a part of TCP/IP which allows a user of a TCP/IP host to request or send files from other TCP/IP hosts.

G8BPQ Code

John Wiseman, G8BPQ, developed a TSR (terminate stay resident) program for the IBM PC and compatibles that would imitate TheNET and allow node access for a program that runs on the PC. This program simulates the TheNET node functionality and allows routing from a TheNET system directly to the PBBS or other program running on the PC. Unlike a TheNET node which can only handle one radio per TNC, the G8BPQ program may direct traffic in and out of several radios by using KISS TNCs or other TNC/modem cards. (See KISS, Dedicated port, Locked node)

Gateway
A configuration of nodes where connectability is available by deliberate manipulation but where automatic end-to-end routing is not possible. This is useful for connecting two networks together such that users and servers on one network can access users and servers on the other network without compromising networking practices on either network.

Examples:

- To access packet radio from Fred's telephone packet gateway I can phone up and use a password. After Fred's machine accepts the password I can use my callsign on Fred's PC.
- To gateway into TCP/IP in Rochester from Albany I can use the TheNET network by connecting into IPROCH which is a PC running NOS. Once I get connected I can use the TELNET program to access another TCPer. IPROCH:WZ2B is a gateway.

HexiPus™
Six way diode matrix card: This is a product of the North East Digital Association (developed by WB2JLR) that allows up to six TNCs to communicate via RS-232. This is used in TheNET and ROSE multiport nodes such that up to 6 radios may be installed at a single network node site. More than six radios/TNCs may be used by adding more than one HexiPus™ and by using a wireline link based on a TNC at each HexiPus™. (See wireline link, diode matrix)

HTS
Hidden Transmitter Syndrome: This describes a condition where throughput is drastically reduced to well below the specified baud rate because a single station is able to hear two or more stations that can't hear each other. (See throughput)

HTS free or HTF
By making sure that every radio/TNC on a frequency can hear every other radio/TNC most of the collision problems and inherent loss of throughput may be removed. At this point backoff becomes effective and the performance of the system of radio/TNCs may be predicted more accurately. The only remaining problems occur when radio dead time due to slow transmit receive switching is excessive. Also backoff must be used if there are more than two radio/TNC sets on frequency. (See backoff)

Internet
The Internet is a public system of computers which communicate over commercial lines (usually telephone or leased telephone lines) using TCP/IP. Usage of the Internet network is free. Usage of the computers that are hooked to the Internet is not necessarily free. Most people who have access to the Internet either pay a fee or have connection to the network from work or school. There is a book on the subject called *Internetworking with TCP/IP* by Douglas Comer and published by Prentice Hall. You should read this if you are interested in the details.

KISS
Keep It Simple Stupid: In the packet world this usually refers to a program or mode that relates to TNCs in which the functionality of a TNC is entirely remote controlled by an external PC or other host. KISS mode is used by G8BPQ code and TCP/IP.

LAN
Local Area Network: NEDA defines a LAN as a user access node and a group of users. Servers do not communicate with the network on the LAN frequency but use dedicated access frequencies. LAN users which are home stations running minimum antenna and power configurations to access the node may access multiple servers through the network via the local access node.

Example:

- A sample network node setup would include a user port that can hear twenty or less active packeteers. No other user ports and no servers share the frequency within the range of the user port radio. The node setup must included one or more backbone links to other nodes. If any servers exist in the area that need network access then dedicated link radios and TNCs are added to the node stack. Users may access those servers (if any user services are supplied) from the LAN frequency by using TheNET, ROSE or other networking technology.

Locked Node
TheNET nodes have the capacity to generate routing lists automatically based on parameters set in the node's RAM. The parameters specify default quality values to be assigned to routes to each neighbor, separately defined for radio port neighbors and RS-232 port neighbors. If a neighbor tells me (I'm a node now) about a certain node, by callsign and node name and quality, I'll remember it for a duration that is also setable by the parameters in RAM. If that duration ends and I haven't gotten a refresh on that information, I'll forget about the node. A locked node is where an individual node is given a specific quality and nearest neighbor route, but for which the duration is set to infinite. This locked node must be manually entered by a sysop but is visible to anybody who wants to look for it by doing a N NODENAME for any nodes that are suspected as being locked.

Example:

- In the NEDA network the default qualities for a user port are 230 for over the RS-232 and 50 for over the radio. The minimum quality to broadcast is also set to 50. In this case any station heard over the user port is listed at the user port but is not delivered over the backbone to other user ports. In the case of a UHF or 220MHz user port there might be servers that are TheNET compatible that should be shown at other user ports in the region. Rather than set the defaults so that all nodes seen over the radio from a specific user port would be broadcast it is possible to 'lock in' a specific node at a higher value. This is routinely done for G8BPQ BBSs in the NEDA network.
- Another NEDA example: if a node performs a special function at a node site or in a certain region it may be locked at a low value so that it doesn't propagate any further than necessary. This is the case with CROWD nodes, DxCluster access ports and BBS/G8BPQ nodes. Each of these special purpose nodes must be visible at all node sites within the designated geographic areas but need not be spread to other areas.

Locked Route

TheNET nodes have the capacity to generate routing lists automatically based on parameters set in the node's RAM. The parameters specify default quality values to be assigned to routes to each neighbor, separately defined for radio port neighbors and RS-232 port neighbors. If I (I'm a node now) hear a nodes broadcast from a new neighbor, I'll add a route to my routes table according to the default quality value in the parameters table in RAM. So long as I keep hearing nodes broadcasts from the neighbor I'll keep the route listed. However, if I hear a node that is not reliable I might decided that the unreliable node is the preferred route to certain destinations. This is called a false route. One possible solution is to lock the route to the unreliable node to 0. Another solution is to lock the desired routes to the required value and then set the default to 0. Or perhaps the unreliable routes or the parameter default could be set to a value that is not 0 but that is such that the false route won't get used by accident.

Mail Drop

A part of a TNC program that allows messages to be loaded into the TNC and then retrieved from over the air or from the terminal at the TNC. Mail Drop is what AEA calls this function. It is also called PMS or Personal Message System, by PacComm.

Matrix

Matrix = Diode Matrix: The TNCs running ROSE or TheNET network software can communicate to each other over their RS-232 ports. If two TNCs are used at a node site the connections are simple connector to connector wire connections. If more than two TNCs are used a diode matrix is required. (See *HexiPus*™)

Matrix Monitor

Communications between TheNET TNCs via the RS-232 port or over a matrix is not in a textual format that is readable by a dumb terminal or protocol analyzer. A Matrix Monitor is a hardware or software device that can display the data passing across the matrix in a form that is legible and informative. G8BPQ code includes a program that can observe TheNET communications over the matrix. KA2DEW developed a crude single board computer with this capability also but the product was never made reproducible. (Anybody want a good project?)

Multistreaming

This is the process by which a user can connects to several stations at once. (See *Stream*)

NBBSC

NEDA BBS Committee: This is a group that converses on issues dealing with server communications and access across the NEDA network. All persons in this committee are NEDA members and are volunteers. The chairman of the committee is appointed by the NEDA Board of Directors. Among other regular projects that NBBSC does is the generation of the server list in the NEDA Quarterly. To contact the NBBSC send a packet message to NEDA @ W1NY with attn: NBBSC in the title field.

NBOD

NEDA Board Of Directors: This is an elected body of 6 NEDA voting members that is the head of the North East Digital Association. The board members are elected for 2 years terms, 3 elected each year, and serve the club. To contact the board of directors you should send a packet to NEDA @ W1NY with attn: NBOD in the title field.

NEDA

The North East Digital Association. This is a club that was formed in the fall of 1989 to support and instigate packet network development. The contact address is Box 563 Manchester NH 03105 or via packet is NEDA @ W1NY.

Neighbor

In a network of nodes the neighbor of a node is any node that is talked to directly.

Example:

- If the linked system consists of FRED <-> BOB <-> ED <-> LEFT <-> RIGHT then the neighbors of ED are BOB and LEFT

NESAC

NEDA Emergency Services Advisory Committee: This is a group that is responsible for interfacing between NEDA and the various state, federal and local agencies (including ARES and RACES) that would be able to make use of the NEDA network for tests or during actual emergencies. To contact NESAC send a packet to NEDA @ W1NY with attn: NESAC in the title field, or send a letter to the club POBox.

Node

A node is an active element in a network. This can mean anything from a user station to a bulletin board. Traditionally a node in packet radio is an intelligent router of real time data, somewhat more intelligent than a digipeater but faster than a store and forward BBS.

NRS

Network Regional Sysop: During the third through sixth quarters after NEDA was formed the NRS was a network conformity tool. The NRS would keep track of all of the nodes in his/her area and make sure that parameters were followed. This was the solution that the board of directors had chosen to network configuration and administration. In the Fall of '91 this system was eliminated.

NTECH

NEDA Technical Committee: This is a group that is responsible for maintaining the NEDA network software and for making recommendations to the board of directors for changes in club and network policy in regards to network technology. The chairman of the technical committee is appointed by the board of directors. The remainder are volunteers. The only restriction on membership are that all members must have direct connect access to the NEDA network at least some of the time and must be approved by the chairman of the technical committee.

Octopus

This is a product of John Painter (rights now owned by NEDA) that was an 8 port diode matrix card to couple TNC2 compatible TNCs to make a multiport node. This product has been outdated by the HexiPus™. (See HexiPus™, Diode matrix)

Packet

A packet is a block of many characters (or bytes) which are sent together along with a few extra characters (checksum) used to guarantee that the data is completely error free. The packet includes addressing information so that the receiving station knows that the packet is for it as well as who sent the packet.

Path

This word is used to mean the nodes, digis and servers that must be used to pass data from one point to another. Sometimes the path may be specified without including some intermediate nodes if the knowledge of those nodes is not necessary to pass the data or make a connection.

PBBS

Either Personal Bulletin Board System or Packet Bulletin Board System. The former is called personal mail drop or personal mail system (PMS) to avoid confusion. Both of these indicate a mail box that is contained inside a normal user TNC.

PMS

Personal Mail System. This is a program that resides in a normal TNC. It usually is included with the TNC as a standard feature when it is bought. The program allows the user of the station, or hams connecting over the radio to leave mail that can be picked up either locally or remotely. Some incorporate the ability to reverse forward. (See forward, reverse forward)

Poll Packet

In the latest version of AX.25 packet protocol if a transmitted information packet is not acknowledged the transmitting TNC will generate a poll packet to see if the destination TNC is still around. If the poll packet is acknowledged then the transmitting TNC will once again attempt to send the information packet. Note that if there is a periodic noise at the receive TNC that the poll packets might be received but that a particularly long information packet might never get through. In that case the retry process might take place until manual intervention occurs. (See retry)

Port

TNCs are two port devices, RS-232 and radio. With network software, like ROSE or TheNET, they can communicate with other network nodes via either port. If a stack of TNCs is networked using a diode matrix then what is referred to as the ports are usually the radio port of each TNC. Thus a 3 radio node is referred to as a 3 port node.

Protected Backbone

A protected backbone is a backbone where none of the known devices involved in the backbone will accept traffic from any unknown device. (See Backbone)

RAM

Random Access Memory. This is an IC in a computer that holds data only so long as power is applied. This is usually used only for storage during the execution of a program. TNCs use RAM for temporary storage, messages and parameters. Normally TNC RAMs are powered all the time using a lithium battery in the TNC.

Real Time

When a signal is sent and a result is expected back within a short enough time to fall within a person's attention span the operation is said to be in *Real Time*. Keyboard to keyboard operation is real time. Keyboard to server is real time. Reading your mail from a remote BBS is real time. Sending a message to a friend via a packet BBS is not real time because the sender doesn't know how long it will be before the friend's answer comes back.

Redundancy Path

In a mature packet network more than one path would exist between every two points. If one of the two paths is preferred due to load handling capability or number of node hops then that path would be called the primary path. The other path would be called the redundancy path. (See Path)

Response Time

This is the time between sending data to a remote device before an expected response returns to the originating station.

RETRY

This is a process by which a packet that is not acknowledged is regenerated.

Reverse Forward

This is a feature of all modern BBSs and some PMSs where a connecting BBS may request if any mail is available to be taken by the connecting BBS. The prompt and answer exchange that takes place is in plain text and may be monitored if you are curious.

ROSE

RATS Open System Environment: This is a networking software package created by Tom Moulton, W2VY, in concert with the Radio Amateur Telecommunications Society in New Jersey which implements a multiport, multistation packet radio network. The software consists of several parts. The most major part is in the form of an EPROM that resides in a TNC2 compatible TNC. Other parts include network management and system operation tools that run on a PC compatible but which are not integral to the network's day to day operation.
ROSE operation is done with the use of site callsigns and numeric designations that are traditionally in the form of a telephone area code and local code. The user treats the local ROSE 'switch' as a digipeater with the destination switch's numeric code as a second digipeater in the user connect sequence.

Example:

- To connect from WB2DWD's station in Long Valley NJ to KA2DEW's station in Potsdam NY a user would do:
  C KA2DEW v NX2P-3,315268
  where KA2DEW is the destination user's callsign, NX2P-3 is the user's local switch, and 315268 is the designation for the switch in KA2DEW's area and on the frequency that KA2DEW will be operating. Note that KA2DEW will get a connect message that looks like:
  *** Connected to WB2DWD via K2CC-3,201876
  where WB2DWD is the originating station, K2CC-3 is KA2DEW's local switch and 201876 is the numeric designation for WB2DWD's local switch. 201876 is the same switch as NX2P-3 and 315268 is the same switch as K2CC-3

The linking methodology between ROSE switches is very open to the design requirements of the implementation. ROSE switches may be linked with a common trunk frequency, with hidden transmitter free backbones, or on the user access frequencies. ROSE switches may be built with from one to many TNCs on many frequencies.
The routing methodology used in ROSE is based on a fixed table stored in RAM in each switch and downloaded by the designated sysop. This may be done locally or from the far end of the network. The routing may list individual switches and include the neighbor for each individual switch or the switches may be listed by class allowing whole 'area codes' to be listed with the same neighbor node. Alternate routes may also be specified. If a link fails completely or is taken off the air the system would adapt quickly.
This software has been in Beta test station for several years and as of August of 90 has been used successfully for building multiport nodes.
A notable difference between ROSE and TheNET is that with ROSE a user doesn't have to know about any intervening hardware between his entry and exit ports in the network. On the other hand the user is also unable to *find out* anything about the intervening hardware.

Route Stepping

One of the features of TheNET is that an individual may hunt through a TheNET networking and take advantage of local Routes commands to determine what all of the neighbors of a particular node are. With this knowledge the user may then connect to a neighbor node and repeat the process. In this way an individual user may entirely map the existing network or collection of networks. This has been taken as a disadvantage by some network developers due to the (apparently) vast amount of network traffic that is generated by this user-play process.

One advantage of route stepping is that if there is a site that is accessible from one end of a network but that is not known on the other side of the TheNET network the user may simply connect from his/her origin user port to a node that knows of the desired site and then connect to that desired site. This may be done repetitively to get to a very distant node.

Another advantage of route stepping is that there are timers internal to the TheNET node that specify how long a packet may take to get from one end of the network to the other. If the packet takes longer than specified by the timers then network-end-to-end retries are performed. This results in unnecessary network load. Furthermore if the retries ever fail then the user is disconnected. By making smaller steps the user may create a more robust path. (See TheNET)

RS-232

RS-232-C is the Electronics Industry Association (EIA) standard number for the most common interface used between computers. This is usually called RS-232. A signal which uses the RS-232 standard is often said to be *at RS-232*. The computer to TNC connection is at RS-232. Normal computer internal data signals use ground and +5 volts to indicate a zero or a one. RS-232 generates -12 volts and +12 volts to indicate a Mark or a Space which are analogs to zero and one. A RS-232 receiver detects a Space as anything greater than +3 and a Mark as anything less than +3. (Note: Your editor cannot verify this information at this time. Please correct me if you can) The reason that this method is used for computer to computer signalling is that TTL (0v -> +5v) is more subject to line noise, capacitance and non-true grounds than is RS-232.

Serial Port & Serial Communications

The part of a computer responsible for sending binary data in a serial fashion. Normally computers talk internally with parallel data signals, that is that all of the important bits for a block of information are sent at once. A serial communications uses only one wire which is toggled many times for a single block of information. Thus a letter A might be sent in parallel all at once when it must be sent as a string of ones and zeros in sequence in serial. The serial port usually consists of a single chip called a UART, a RS-232 driver chip and a connector.

Server

A server is any stations that is operating as a service to other users than the owner. This may included BBSs, DxClusters, DOSgates, TheNET nodes, ROSE nodes, TCP/IP hosts etc.

Site Hardening

Term for ruggedizing a site by adding backup power, shielding or lightning protection. This includes weather protection and protection against RF problems or nuclear attack.

Site Manager

This is the person or persons that are responsible for node hardware and site access.

Site Sponsor

This is the person or persons who are financially involved in acquiring and maintaining node hardware.

Site Sysop

This is the person or persons who have software control over a node site.

SMTP

Simple Mail Transfer Protocol. This is the part of the TCP/IP system which is responsible for sending mail between TCP/IP hosts. This is a non real time service which operates in a way very similar to normal packet BBSs. Mail is generated by a user at a TCP host, with a destination address. The host makes the decision of what other TCP host the mail should be routed to to get to the specified destination. Eventually, in zero or more hops the mail gets to the destination host.

SSID

Secondary Station IDentification. A callsign is normally used as an address. In a case where a ham needs to have more than one address on the air at the same time the callsign may be used with an ssid. There are 16 different possible SSIDs. NK1M-2 is an example of an address using a callsign and an ssid of 2. NK1M is the same as NK1M-0. -15 is the highest ssid that can be used.

A function of TheNET, G8BPQ, MSYS and other node software is to change the ssid of a callsign that passes through the node or network of nodes. If a station connects to a node with an ssid of 0, when the station connects out of the node with an ssid of 15. The formula used is 15 - ssid = exit ssid. Thus a station using an ssid of 4 will emerge from the node or network of nodes with an ssid of -11.

Stream

AX.25 allows many connections to be made from one station at the same time. Each connection is called a stream. The origin and address callsign pair for the connections must be different for each stream. That is: If I am KA2EIA and I connect to three other stations, I can connect to NK1M, K1MEA and NQ1C but I cannot connect to NQ1C, NQ1C and K1MEA. This process is called multistreaming and is available on most modern TNCs. Look at the *USERS* command in your TNC's manual.

TCP/IP

Transmission Control Protocol/Internet Protocol. This is a suite of protocols that define the operation over the 'Internet'. This package of protocols was used by Phil Karn, KA9Q, for the creation of a packet radio version of TCP/IP. As this is a fairly mature networking system it supports many features not available in the current 'made for ham radio' protocols. It also has features that would take much better advantage of networking resources for the transmission of volume data than do TheNET and ROSE. One problem that TCP/IP for ham radio currently has, however, is that it requires a more sophisticated computer and a more sophisticated operator than are required to operate ROSE and TheNET. See also *Internet*.

TheNET

This is a networking software package created by Hans Giese, DF2AU, in concert with NORD><LINK in Germany which implements a multiport, multistation packet radio network. The software consists of several parts. The most major part is in the form of an EPROM that resides in a TNC2 compatible TNC. Other parts include node configuration tools that run on a PC compatible but which are not required after initial system setup.

TheNET operation is done with the use of site callsigns and mnemonic designations that are traditionally in the form of a city name. The user treats the local TheNET node as a remote command processor by first connecting to it, then away from it.

Example:

- To connect from WB2DWD's station in Long Valley NJ to N2MGI's station in Potsdam NY a user would do:
  C SUSSEX, then when connected,
  C POTSDM, then when connected,
  C N2MGI where N2MGI is the destination user's callsign, SUSSEX is the user's local switch, and POTSDM is the designation for the switch in KA2DEW's area and on the frequency that KA2DEW will be operating. *This would only work if POTSDM showed at SUSSEX's nodes list. In practice with TheNET each connect step can only be a few node steps.*
  Note that N2MGI will get a connect message that looks like:
  *** Connected to WB2DWD-15
  where WB2DWD-0 is the originating station

The linking methodology between TheNET nodes is very open to the design requirements of the implementation. TheNET switches may be linked with a common trunk frequency, with hidden transmitter free backbones, or on the user access frequencies. TheNET switches may be built with from one to many TNCs on many frequencies.

The routing methodology used in TheNET is based on a dynamic table stored in RAM in each switch and automatically generated by periodic information transfers between nodes and within restrictions placed on each TNC by the designated sysop. This may be done locally or from the far end of the network. The routing lists individual nodes and includes the neighbor for each individual node. Alternate routes are automatically generated but in practice are not used. If a link fails completely or is taken off the air the system will adapt to the lack of that link after a number of hours.

TheNET has been in use now for several years in the US. Recently NJ7P, Bill Beech in Arizona has been releasing heavily modified versions of TheNET under the name of TheNET Plus.

A notable difference between TheNET and ROSE is that with TheNET a user can delve into the routing tables of each of the nodes and find out how the network is put together. The user can also determine from available information at each node TNC how well the node is managed and how well it is integrated into the surrounding network equipment. On the other hand the user is *required* to know at least some information about the network's architecture, at the minimum, and in some areas the user needs to have a very complete knowledge of the architecture in order to use the TheNET nodes effectively.

(See also *ssid*.)

TheNET PARMS

TheNET nodes, which run in TNCs, operate using timers and other parameters that are initialized in the EPROM when it is burned. Some of these timers and parameters may be modified over the air by the site sysop. A complete description of these parms was published in the NEDA Quarterly volume 2 number 2. This info will be reprinted in the 1992 membership package. (See TheNET, NTECH)

Three Way Wireline Link

This is a circuit that allows up to 3 TNCs to be connected together as if they were over the air to each other. This circuit bypasses the modems in each TNC so that the three TNCs may communicate at high speed. The three way wireline link circuit was presented in the NEDA Quarterly volume 1, number 4 and is also presented in the current NEDA *Annual* membership package. (See wireline link)

Throughput

This term specifies how many bytes sent by an origin station actually reach the destination station in a give period of time. Throughput is a much better number to describe network performance than baud rate. Baud rate only describes the number of bit transitions that may possibly leave a transmitter in a second. Throughput is a statistic that actually means something to an end user. Throughput is calculated either by observation or by taking the original baud rate, given in bytes per second, subtracting out all of the wasted time and overhead due to network protocols, the lost time due to choking and due to collisions. (See Choke)

TINK

Slang for TNC.

TNC

Terminal Node Controller. This is the brains that connects the user's terminal to his radio so that he can communicate to other stations. The TNC's job is to take text typed on the terminal or computer and store it until the user hits a return. At that time the text is sent to the destination station. Each line of text (ended with a carriage return) becomes a packet and is stored in the TNC until it can be sent to the destination station.

The TNC also has commands that let the user set the callsign of the station and set up communications with another station or stations (Connect command).

The TNC is like a home phone modem in that it converts digital character data to tones. The big difference between a TNC and a phone modem is that the TNC has the intelligence to direct your traffic to a specific destination and to receive connects using it's own microprocessor and internal software. A phone modem is relatively stupid in that it can only work on a channel where there is only one destination, i.e. a telephone.

By changing the internal software TNCs may also be used for other purposes besides home station use. This includes running as part of a set of TNCs in a network node.

TTL

Transistor Transistor Logic. This is the name for logic circuits which operate using 0V for a zero and 5V for a one to do binary operations.

UART

Universal Asynchronous Receiver/Transmitter. This is an IC which is used in a computer to construct a serial port.

User Channel

The frequency designated for users to access the network. This frequency would be devoid of servers and other nodes aside from the one designated. (See LAN, WAN)

WAN

Wide Area Network: This is a system where many servers and nodes may talk to each other. This kind of system is rugged in that communications would probably not be compromised if a single site went off the air. The major problem with this methodology is that if the only packet systems available are of this type then users, which present transient loading, will find that the WAN is unable to support massive intermittent loads during peak usage times. This causes frustration and leads to non-utilization of packet.

Example:

- 145.01, 145.03, 221.11 and 441.0 are commonly used in this configuration.

Wireline Link

This is a connection between the modem headers of a pair or more of TNCs such that the TNCs communicate via their radio ports but without an intervening pair of radios. Because the modems are bypassed the TNCs may talk at a higher data rate than 1200. (See Three way wireline link)

Wink 'N Blink Mod

This is a modification to a TNC2 that allows that the CON and STA lights are used to monitor the RS-232 port's DCD and CTS signals. These signals act as PTT and Carrier Detect on the RS-232 so making this mod allows an observer to watch activity between the several TNCs at a single node site. This is an excellent diagnostic tool and is fun to watch. This circuit was described in the Fall NEDA Quarterly of 1990, volume 1, number 4 and is also described in the current NEDA *Annual* membership package.

NEDA Quarterly
Compendium of past issues

The following pages have all of the technical articles printed in past Quarterlies that are not specifically outdated. Some of the information is redundant with what is published earlier in this Annual. They are still included so as to show history and differing points of view. Some also deliver the information in different ways. This may help with explanations for topics which are not altogether clear.

Also included are all of the graphical articles for packet node sites that have been in past Quarterlies. Although the sites involved may have changed and the information is no longer current they still stand as decent examples of network construction.

Disclaimer: Unless an article is specifically marked as representing NEDA or the board of directors or another officer the articles presented in this section are from the point of view of the author. They do not necessarily reflect NEDA policy. Most of the material in this section does, in fact, represent the feelings of the membership. Please keep an open mind. If you'd like to submit an article to the Quarterly or enter a letter to the editor, send to the editor's address, the PO Box or NEDA @ W1NY.ma.

Table of Contents

Improving switching performance of the Wilson MH400

Copied from NEDA Quarterly #1.1

The Wilson MH400 walkie talkie was the first of the NEDA group purchases. Dana, WA2WNI, orchestrated the buy and around 35 of the units are now in the hands of NEDA members.

The original application of the radio was in the UTICA -> utica2/ frankfort backbone link. Howie, WA2TVE, and friends were able to get the radios to work in the 430 MHz region and set up a full duplex link. On the basis of this success the radios were ordered in mass quantities. Unfortunately one of the things that was never tested was the Rx-Tx-Rx turn around time as Howie's system was full duplex! Alas when Linds, (NR1N) got his two units crystalled up he found an incredible delay. He had to set his TXDelay to 50 to make the nodes work!

Well.. Herb, WA1TPP, who owned a couple of the units, decided that this just wouldn't do and proceeded to generate the following modification:

Purpose, procedure #1: To modify the Wilson MH400 HT so that the Rx-Tx-Tx time is decreased.

Description, procedure #1: Note in the lower right hand corner of the schematic for the MH400 switching transistor Q704. Q704 works with regulator Q705 and controls voltage to the transmitter section of the radio. Notice that the base of Q704 has an R/C combination between it and the PTT line. The time constant of this R/C circuit (C706 + R710) affects the key up time in the radio. We will eliminate this delay by breaking the circuit at R710 (10K).

Procedure #1: Remove radio from its case and expose the under board as you would to install xtals. As you view the under board with the ant pointed away from you locate variable resistor RV701 (lower right center). Looking to the left of RV701 notice C701 then Q705 then glass diodes D706 and D707. Directly above Diode D706 & D707 notice resistor R710. This resistor is end mounted and is 10K ohms. With a pair of small cutters carefully cut the bent lead coming from the top of this resistor.

Note: The schematic also shows a similar R/C combination on the base of Q701. This R/C circuit (C702 & R705) *was* not installed in my radio. If it were it would also *increase* Tx-Rx turn-around time.

Purpose, procedure #2: audio recovery timing improvement.

Description, procedure #2: To demonstrate the problem turn the radio on, release the squelch, key and then unkey the radio. You will notice a delay between the time the radio is unkeyed and audio reappears. The purpose of this mod is to remove this delay.

Look in the upper right hand corner of the radio's schematic. Notice Q602 (squelch switch). Right below it notice two diodes D601 & D602. Notice their cathodes connected to capacitor C612 (2.2 μfd). This capacitor causes the squelch delay described earlier. The solution is to remove it.

Procedure #2: As in the first modification expose the back board and locate variable resistor RV701. Notice directly below it C612. You can remove this cap by unsoldering it but I preferred to rock it *gently* until the leads fracture and not risk soldering iron damage to the board. When C612 is removed you will notice that the time delay (Unkey --> return of audio) mentioned earlier is gone.

Purpose, procedure #3: Improvement of Tx-> Rx switching speed.

Description, procedure #3: After you have performed the first modification to the MH400 you will have noticed a dramatic increase in Tx-> Rx switching speed. However, there is one further improvement you can make. Notice in the lower right hand corner of your schematic Transmit Switch and voltage Regulator Q705. Notice the R/C pair connected to its base formed by RV701 and C708 (2.2 μfd). R714, R713 and TH701 also play a part in establishing the time constant for the base of Q705. This time constant will determine the time difference between Key Down and RF Output.

Procedure #3: Adjust RV701 (output power control) for full power. I found that on my radio doing this made a big difference in the time between Key down and RF out. This adjustment will alter the time constant on the base of Q705. (Hint: RV701 has no rotational stop. Use a watt meter or carefully observe the wiper when adjusting it for full power output - full clockwise)

If things are still moving too slow or if you want to run at reduced power you may remove C708. C708 is located just above RV701 on the back board and can be removed in the same way C612 was as described in a previous procedure.

—Herb Belin, WA1TPP

RFCARC Landline BBS

Copied from NEDA Quarterly #1.2

The RF Communications ARC (WB2PSI) maintains a Land Line BBS with an extensive collection of software of interest to Hams. Called "the Vector Board", A-L-L types of computers are supported, something a bit unique amongst most BBS's. Over 10M of Packet related software is available including the latest releases of W0RLI, WA7MBL, AA4RE, MSYS, NOS, and more. Also available are 40M of EE and other Ham software covering subjects like Circuit Design, Propagation, Antennas, OSCAR, Logging Aids, CAD, and much more! Operates 24hrs/day, open access, donations welcome! Call 716-544-1863 in Rochester, NY 300 through 9600bps MNP/USR supported.

VHF Cluster Proposal

Copied from NEDA Quarterly #1.2

Here in the Rochester area there is a fair amount of weak signal VHF activity. One problems is that no one knows when the bands are going to open, hence they'll probably miss some much needed grids unless monitoring all the time. In the Rochester VHF group there used to be a system where stations gave one another a "one ringer" on the phone as a band opening alert. There is now a simplex 2 meter frequency used for this purpose, however many stations are outside the 2 meter simplex range. This sounds to me like a perfect application of packet radio. What is needed is a VHFCLuster similar to the Dx-Cluster systems now in use for HF DXing. Since no such VHFCLuster system exists yet, I propose that one channel on one or more CROWD ports be designated as a weak signal VHF spotting channel. Aurora or E skip openings would no longer be missed due to beam heading or lack of monitoring. The CROWD would also make an easy way to set up microwave schedules with other stations and stations could also advertise what grids they are looking for on which bands etc. If we try and use too many different CROWDs to start with, there may not be enough stations on any one CROWD to make it useful. To get things rolling I'd like to suggest that we use channel 144 of the CROWD at BERK and the one at CANDGA. Please think about this idea and send me your comments or bring them to the July meeting. Maybe we can come up with some more formal decision of whether to adopt such a plan. Should we set aside any other CROWD channels for special purposes (like Emergency traffic for example)?

—Rich Place, WB2JLR

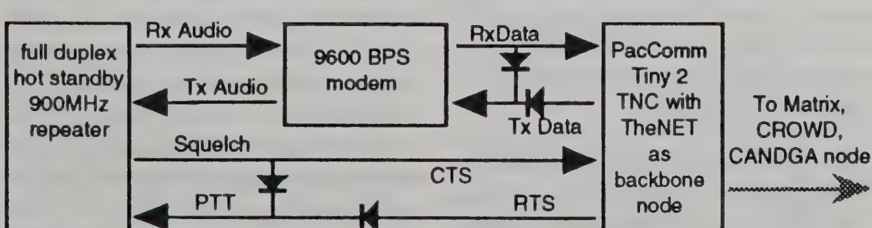
Need some NEDA publicity materials? Going to a flea market? Contact any of the board of directors. We have a 4 page flyer that we've been cranking out literally by the thousands.

Feel free to photocopy maps and other NEDA materials, just give NEDA credit.

9600 Baud 900 MHz Repeater

Copied from NEDA Quarterly #1.2

The CANDGA node in Canandaigua NY is now using a 900 MHz full duplex repeater which includes and internal 9600 baud modem. The equipment, donated by Microwave Data Systems in Rochester NY, will provide connectability from MONROE in Rochester to COR144 in Cortland, over 80 miles away. Being full duplex completely avoids the hidden transmitter syndrome since anyone within range of the repeater will hear anyone else transmitting to the repeater. Data being sent between MONROE and COR144 will bypass the CANDGA TNCs, going out of and directly back into the modem at the repeater. This speeds things up an additional factor of 2 over a conventional store-and-forward type node. If the packets are addressed to CANDGA, then it intercepts them and responds. Here is the configuration for anyone else thinking about doing something similar. Rich WB2JLR @ KC3BQ



Corrupt Data

Copied from NEDA Quarterly #1.2

About a year ago when the network was "new" we had a problem with "corrupt" data getting into the nodes broadcasts from some location out east it was believed. The problem manifested itself by not allowing anyone to do a NODES list because as soon as someone did this their stream into that nodes command processor would crash (via immediate disconnect). This was just *one* of the reasons that led to careful control of what entered the network via *user* ports.

The problem seems to have occurred, but I have made an interesting discovery.

Version 1.1 of TheNET is *not effected* by this perhaps due to its ability to not recognize #pound nodes and such, but only valid *callsigns*! I have a Version 1.0 chip here for my *wireline* link, and it blew itself up (crashed) every time I asked it for a NODES list which contained the corrupted data that was being passed around on the nodes lists. The other ports at my node are all version 1.1 and they did *not* crash on request for a NODES list, but gave a corrupted list that showed a node that looked like L:^D^D^B^D^D^E-2 Unfortunately, the wildcat information was not readily traceable to determine just where it occurred. I have several recommendations. 1) this is a *good* reason why we should *upgrade* all nodes to version 1.1 as soon as possible. They are not affected by this known bug. 2) If this should occur again, sysops should *isolate* their section of the network by *stopping nodes broadcasts* between regions and then *looking* for the source of the bad data.

This could provide useful insight to what is occurring and possible countermeasures to insure such stuff doesn't clutter our nodes lists again.

Any further data regarding this phenomenon should be passed on to Tadd.

—Dana Jonas, WA2WNI

Death By Competition

"I know of no safer depository of the ultimate powers of society but the people themselves. And if we think them not enlightened enough to exercise their control with a wholesome discretion, the remedy is not to take it from them but to inform their discretion by education."

Thomas Jefferson
Co founder, United States of America

Before reading the rest of this editorial comment I would like readers to re-read the above statement, and forgetting about ourselves as amateurs consider how our country would have developed if not for the framework of our government which was forged by men like Thomas Jefferson.

It is no wonder that our country developed as fast as it did for information on what we had, how good it was and what made things tick was readily available to anyone who cared to know. Furthermore anyone who wanted to be a part of it all had only to express sufficient interest, intellect and time for involvement to be immediately planted firmly in some aspect of the grand scheme of things. No aspect was truly closed to those desiring involvement. Politics, society, industry, administration, agriculture, exploration and dozens of other areas were attractions for anyone with such leanings.

I would like everyone to consider that Amateur Radio is just as open and just as ripe for development and involvement as the example above. The problem is that without this understanding and an exchange of free information it cannot continue to attract involvement by those who are best capable of helping us grow. If this happens we will stagnate and wither in our knowledge base thus creating our own demise.

So what is the bottom line here?

Involvement
Information
Cooperation

It is hard to believe that anyone reading this who is a NEDA member is *not* already involved in packet to the point that they can actively promote our cause. And *What* you ask is our cause? Very simply it is to implement, promote and widely share those technical advances in networking and implementation philosophy that ultimately improve interconnectability of *all* packet services. Not just users. Not just Bulletin Boards. *Not* just special services like a DOSgate or DxCluster or CROWD port and *certainly not* just any *one network* containing such things within it! *All packet services* - inclusive - in its entirety - everywhere - globally. We must openly encourage the *wholesome discretion* referred to by Mr. Jefferson but also make sure it is an *'informed'* discretion. There is much 'power' to implement our technology with mostly just the use of the correct information, but there is no better way to efficiently use technical resources and hardware than to universally share whatever is available for all applications. This does require up front planning when doing things however as retrofits and add-ons rarely (if ever) achieve such efficiency.

Death by competition

The real killer of efficiency is non compatible independent implementation of redundant applications. Yes, we see them all the time in the private sector because in the commercial world most communications users are in *competition* with each other. They wish to have *independent* paths of communications that each individual group controls totally to their own financial (or image) gain. Certainly this is all right if such individuals can afford it all and it doesn't detract from the capacity of another group to reproduce their own services, but increasingly this is proving not be the case.

A very recognizable example of this effect in the non- ham world recently has been brought to light by the needs of state governments in their statewide communications networks. The problem was that the lack of coordinated government and public safety communications services caused enormous drain on statewide budgets. The statewide law enforcement agencies had a statewide network; the departments of education, transportation, health, social services, fire services, civil defense, administration, etc. ad nauseam each had their *own* statewide communications network! The cost and maintenance of each system was variable because they were all just slightly different, but the overall cost of both implementation and operation was *enormous* not to mention the fact that these systems rarely had capacity to cross communicate! Budget administrators, upon uncovering this "turf protecting" policy creating essentially private (to each agency) networks for each agency, rioted at the misuse and inefficient application of taxpayer's bucks.

Centralized Telecommunications Agencies with the directive to create a single statewide network with more than enough capacity to handle *all* government agency needs were quickly created by state governments capable of fast response and dire need for financial recovery. Previous independent networks were rapidly phased out and integrated network use mandated. Systems created with long term objectives of integrating services quickly proved to be effective *and* drastically reduced the expenses involved with keeping themselves going. Public service agencies discovered the real value of "interoperability". Indeed, several state systems paid for themselves in record time!

How was this done? Cooperation and joint involvement was the key. It is interesting to note that in many states where such cooperative efforts were not mandated by some high level authority the wallowing in financial mire and inefficiency still goes on. This sure says something for cooperation in government eh?

There are really not many of differences in amateur packet networking and government communications services networking. We both work on limited funds, resources, manpower, support, time and a few other things as well (recognition and respect for personal sacrifice for others in the performance of public services for example?)

There are now in existence some real good examples of large scale amateur cooperative efforts that have achieved significant, even historical, events. Some of them have really opened the eyes of government and commercial observers who then copy our examples. New mode developments, super inexpensive communications satellites in orbit serving globally and now wide scale data networking on a flea power budget. Lets get on with things as innovators, supporters and educators of our fellow amateurs. But most of all, lets do it *together*!

—Dana Jonas, WA2WNI
Copied from NEDA Quarterly #1.3

Speedup for Relay Keyed Radios

Copied from NEDA Quarterly #1.3

Here's a simple trick to improve those radios of yours with relay keying. While modifying for PIN diode switching would be great, some radios switch so many different voltages that it's not feasible to go to solid state switching. I tested several relays and found that while most will turn on in under 10 msec, the time drop out time was at least 3 times that long. When the relay is turned off, current continues to flow through it via the snubber diode. The current dies out exponentially depending upon the relay coil's resistance and inductance, but in the meantime it keeps the relay energized. By adding resistance in series with the snubber diode you can decrease that time constant and speed things up. Adding this resistance will increase the voltage transient that the keying transistor will see, so you must be careful not to use too big of a resistor. The procedure to follow is this:

>Look up the Collector - Emitter voltage rating of the keying transistor hooked to the relay.

>Divide the voltage rating by the voltage applied to the relay and write down that ratio.

>Measure the resistance of the relay (measure it both directions and use the higher reading so that you aren't just reading the diode across it.)

>Multiply the resistance you measured by the ratio calculated above, and round that number down to the nearest available resistor value.

>Add this resistor in series with the snubber diode that is across the relay.

When you are done, the voltage transient that occurs at the collector of the transistor when it unkeys will be just under the voltage rating of the transistor. The time for the relay to unkey will be reduced by a factor equal to the ratio calculated above.

—Rich Place, WB2JLR

AT LAST: Relief for Ailing 13-509 relays!

Copied from NEDA Quarterly #1.3

A new technical expert has emerged in our midst in the form of Mark Oliver NM2J - Burnt Relay Eradicator -

Using the basic construction techniques outlined in a 1988 issue of Ham Radio magazine, Mark has gotten the Midland 220 radio rehabilitation project down to a science. By using several guinea pig radios provided by Dana WA2WNI from the Hudson Division (seriously fried by overactive use on the ALB to STMFRD 220 backbone link earlier this year), the details of what to snip, solder and stuff into the limited space in the rear of the rig were devised. The "renewed" radios will be put back into network service soon.

The backlog of such units waiting for repairs from just the NY portion of the network numbers about 6 units and it is unknown how many more rigs may be in critical need of upgrade.

Mark has indicated he will be glad to perform these valuable upgrades as time permits for NEDA Network site ops owning 13-509 rigs that ARE or WILL BE used in the network for backbone services. The expense of the modification is basically the cost of several PIN diodes, small switching transistors and the small circuit board assembly that handles Tx/Rx B+ switching.

NEDA Network site ops may send further inquiries to NM2J @ WB2WXQ (and don't demand overnight service as Mark has a NODE with a USER port, two dedicated USER ports and two backbone links to take care of too!). Rumor has it that WA2VAM has a whole stack of burnt 13-509s multiplying in the dark corners of his ham shack closet just waiting for some 13-509 mod expert to tackle them. - Look out Mark!

—Dana Jonas, WA2WNI

Wireline Linking of 3 TAPR2 TNCs at 9600 baud

Copied from NEDA Quarterly #1.4

TNCs are two port devices. For TheNET we tie the serial ports of several TNCs together to form a multiport node. For TheNET this works. But what if you want to tie a different kind of device into your TheNET node, like a DOSgate or personal TNC station running normal TNC1.1.7 software or personal mailbox software? Or, what if you want to tie more than 5 TheNET nodes together? The RS-232 drivers won't like this.

Here is a solution. What I've done is connected the modem headers of three TNCs together. This allows the TNCs to talk, as if over the radio, to each other. Because the TNCs talk using AX.25 level 2 they can all talk to each other, even if one is running TheNET, one KISS and one a normal AX.25 user station. Another popular use for this is to tie a pair of TheNET clusters together at the same site while leaving a debugging station, home station or BBS tied in all the time. If you already intend to tie the two TheNET clusters together you can add a user station for the cost of just the user TNC and a couple of TTL chips.

Each TNC2 has a modem disconnect header. This header is hooked in between the Z80 SIO (serial interface chip) and the modem/PTT/DCD circuits. All signals at the disconnect header are at TTL (0v/5v) levels. This means that we can play with these levels using regular cheep logic chips. Another thing this means is that the data travelling along the wires will be susceptible to noise so keep the wires less than a couple of feet long. Just long enough so that the TNCs can be taken apart will work fine.

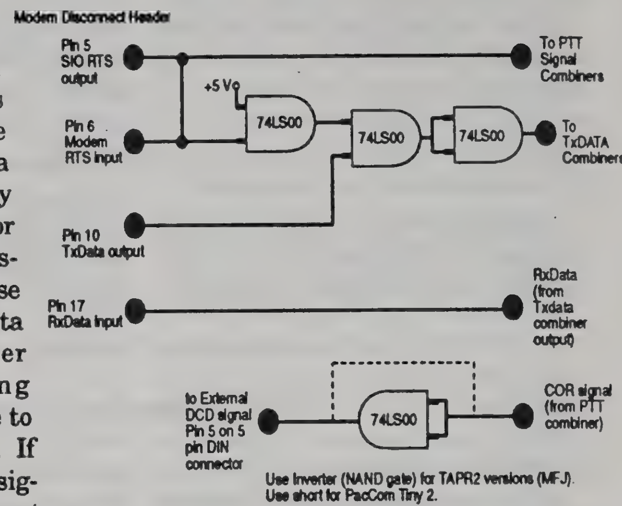
Because we are bypassing the modem circuits we can set the radio port baud rate up as high as it will go. Tiny 2s run at 19.2Kbaud.

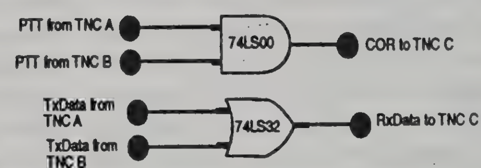
The circuit:

Each TNC had 4 lines which come out of each TNC box. The lines are Tx Data, Rx Data, PTT and COR. The modem header has these signals on it but there are some problems. The Tx data from the SIO is normally gated in the modem and/or by the radio that is transmitting the data. Because we're hooking the Tx data line directly to the other TNCs without going through a radio we have to gate this data ourselves. If you look at the Tx data signal on the modem disconnect header (pin 19) you'll see that it's putting out a square wave all the time the TNC is on. When the TNC goes to transmit the RTS output line (pin 5 of the disc header) goes low. So I have inverted the disc header RTS signal, then we use this signal to gate the Tx data signal. This process inverts it so we use the 3rd part of the 74LS00 to invert it back. Next the Tx data signal is ORed with the Tx data from another TNC and the output of that goes to the 3rd TNC. The Tx data from each TNC goes to 2 OR gates, one for each of the other two TNCs. Your combiner circuit will use 3 OR gates (74LS32) and 3 NAND gates (74LS00). The circuit in each TNC takes 3 or 4 NAND gates. Because there are 4 gates in each package you will need a total of 5 packages to make the 3 TNC coupler. I mounted the perf board which held the 2 coupler chips in the case with one of the TNCs. I mounted each of the TXGATE circuits in the TNC which it served.

The DCD circuits for the Tiny 2 and the MFJ1270 are different;. You'll need an inverter in line with the DCD signal for the Tiny 2 (shown in the diagram). The reason that we run the DCD signal into pin 5 of the DIN radio connector is that this way we get to see the DCD LED work.

The light show is fabulous when this thing is working. For further





light show with the Tiny 2 running TheNET try this modification:

Wink and Blink Mod

Lift pins 16 and 25 on the SIO. (You'll have to pull the chip out of the socket, bend the pins out straight and plug it back in). Now put a jumper between pin 16 of the SIO and 23 of the SIO and another between pin 25 of the SIO and pin 24 of the SIO. Do these jumpers on the bottom of the board.

What this does is make the STA light indicate RS-232 Receive activity and the CON light indicate RS-232 Transmit. If you are using the 3 way wireline to tie two or three TheNET clusters together and if you do the STA/CON modification to each TheNET TNC in the 3 way then you'll have a decent diagnostic tool to show you all of the activity on any of your TheNET ports.

Another trick

If you are coupling two TheNET clusters together and using the third TNC as a user station you can set the third TNC into TRACE mode ON and watch the TheNETese between the two clusters.

—Tadd Torborg, KA2DEW

So What's a Dx Cluster?

Copied from NEDA Quarterly #1.4

A server is any system provided as a service to other users. A BBS is a particular kind of server.

Another kind of server is the Dx Cluster. It's full name is Dx PacketCluster™. This server which is a software product of Pavilion Software is a sort of combination between a BBS and a CROWD port. It is intended primarily for the purpose of HF DXing. Stations share data regarding the spotting of rare DX on various HF bands. Users of the DxClusters typically log into a Cluster server and either observe and respond to "spots" coming in thru the system or participate by finding acceptable "Dx" on whatever HF bands they are tuning around on and then share them with other stations logged onto the DxCluster network by issuing an official "spot".

From the NEDA Network there are DxCluster servers available from a number of *dedicated* user ports. To find one of these servers look for a node name with "Dx" in its mnemonic and connect to that port. Issue the INFO command to find out its location and the last step connect command to get to the actual server off that port.

Examples of these ports include DXCLUS, RDXA, BUFDX and YC-CCDX. It is important to realize that these special servers keep streams of data passing between each other to share the large database of info that all the combined users provide. Thus users accessing these servers should *not* connect to more than one DxCluster station at one time as you will merely be adding more loading to the network unnecessarily. The data that one server has is shared with all the other servers through inter-connected data streams between clusters. It is actually best for the network if users wishing to access DxCluster stations try to connect to the DxCluster server that is closest to them via the network.

These special servers also provide additional services such as Dx country info, QSL bureau info domestic QSL handling, antenna bearing headings for Dx stations, propagation forecasts and limited E-mail storage as well.

—Dana Jonas, WA2WNI

PacketCluster™ is a trademark of Pavilion Software

Announcing the NEDA HexiPus™

Copied from NEDA Quarterly #2.1

The N.E.D.A. HexiPus™ is now ready for delivery. This spin-off from the original Tjp Octopus is a 6 port diode matrix TNC coupler. The PC board for the HexiPus™ is double sided, plated through and includes holes for 60 diodes and 6 nine-pin Dshell connectors. The board is shipped either as a kit with diodes or as a kit with diodes and 6 connectors. The board is 3" by 4", solder masked and silk screened. NEDA has Rich Place, WB2JLR, to thank for the design and leadership in making this product available. Thank you Rich. See the article elsewhere in this Quarterly for details on the history of the HexiPus™/ Octopus.

Prices and order info:

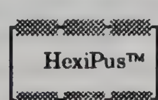
For the diode and board kit

send $22.95 + $4 shipping and handling to the club POBox.

For the kit with connectors

send $29.95 + $4 shipping and handling to the club POBox. No quantity pricing at this time. Expect 4 weeks for delivery.

—NEDA



QSL Information Available Via Packet Radio

Copied from NEDA Quarterly #1.4

The QSLMGR server is a packet radio based service that keeps track of DX stations and their QSL manager or addresses. You may send requests for information to the server, and you will receive a response with the information you requested. If the server does not have the requested info on file, your request will be stored and you will receive a message if the information becomes available.

The QSLMGR server is a unique service because it not only allows you to retrieve information from a database, but it allows you to *update* and *add* to this database - Allowing the whole packet community to keep this information as current as possible.

There are currently over 8,000 entries in the database.

To request information from the server, simply send a message to QSLMGR @ W1NY.MA.USA.NA. In your message, list the calls you are seeking information on, one per line. If you'd like more information on the server and it's various commands, send a message to QSLMGR @ W1NY.MA.USA.NA with the word HELP as the message text.

The server has helped many find their Dx QSL routes - I hope it can help you, too!

—Bob Lafleur, NQ1C

Suggested Packet Settings

Copied from NEDA Quarterly #2.1

Are you having trouble with more retries than you should when the frequency is busy? Or maybe one station is hogging it all and you can't figure out why there is no room for your packets. Here are some ideas that may help the situation. Get together with your fellow packeteers that are operating on the local user port and review the settings of the timers in your TNC's. Much of the following information was contained in a couple of articles by WB6RQN in 73 magazine some time ago, and we have found the suggestions to be very helpful on the VNH user port. It does require a cooperative effort of all the users.

TXDelay

Run it as low as you can and still work the stations you normally converse with. I can run my TXDelay at 8 quite reliably when working VNH. VNH has a very fast receiver, however a setting of 20 to 30 is required to accommodate some of the local stations (I'm working on them, hi) that connect to me in the occasions that they can work me direct and not through VNH.

Note: True DCD in the TNC will drastically improve your receive response time. True DCD boards are available from PacComm for about $26.00 and these will work with most TNC's that have the 3105 modem chip. It's very easy to install, you just pull out the 3105, plug in the board, then plug the 3105 back into the new board. Open up your squelch, and away you go. The MFJ TNC's are represented to have true DCD already installed, but I have found it does not work effectively with some receivers. The only way you can tell is to try running your squelch open, and see if the DCD light shows on receiver noise. If it doesn't light with your audio gain control set properly you've got it made. Running with the squelch open makes for a faster receive system.

Non-Persistent CSMA

If your TNC has 'non-persistent CSMA' (Carrier-Sensed Multiple Access), use the following settings. You can determine this by looking at your commands list. If you have it, the commands PErsist, PPersist and SLottime will be in your DISPlay.

SLottime: Set to equal TXDelay

PErsist: Set to equal 255 x (1/n) where n = the number of stations using the channel other than you.

PPersist: set to ON

DWait: Set to 0

If you do not have non-persistent CSMA then DWait is set to twice the value of the highest TXDelay setting of all the stations in the area. This will do a similar job to what non-persistent CSMA does.

FRack

Set to 5 or greater, 10 works pretty well if it's busy.

MAXframe

Set to 1. This gives everybody's packet a chance at the frequency, in its respective order. Plus it cuts down on retries if the frequency is loaded with hidden transmitters. I have found this one to be a highly controversial issue. 6 people will give you 6 different answers on what is best for MAXframe, plus the square of that number of reasons why one is better than the other. All I can say is 1 works best for me. If the frequency isn't busy and there are no hidden transmitters around hitting your packets with big blasts of data then higher numbers will work fine.

—Cal Stiles, W1JFP
Mount Ascutney Packet Radio Association, (MAPRA), The VNH folks.

Pre/de-emphasis

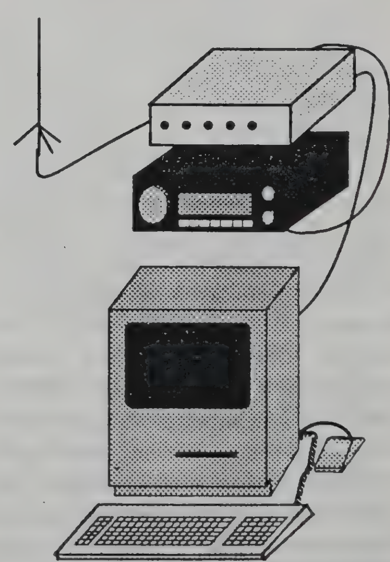
Copied from NEDA Quarterly #2.1

Using 1200 baud modems and FM radios there is a theoretical advantage of not using pre-emphasis and deemphasis. Tests run show the advantage to be in the neighborhood of 2 dB.

Knowing this, and wanting my User port to work its very best, I made a serious mistake; I disabled the pre-emphasis and deemphasis. To get the 2 dB advantage you must disable the emphasis at both ends of the link. Defeating it at one end of the link but not the other results in delay distortion of the data, which can be disastrous. Depending upon the modem chip used, the radios will either work marginally, or not at all. Since most users connect to their TNCs via the microphone and speaker jack, they have pre-emphasis and deemphasis, so the User port radio in the node needs it too.

On the other hand, if you have a long haul backbone link and think that another 2dB will make a difference, this would be an easy way to get it. Just be sure that both ends of the link get the changes.

—Rich Place, WB2JLR



SNH node in Windham New Hampshire

Copied from NEDA Quarterly #2.1

The SHN node has been on the air from K1TR's house since Nov 1986. The station originally came on the air as a plain digipeater thanks to the efforts of KA1MGO and the New England Packet Radio Association. The node was converted to a NET/ROM in fall 1987 and served as a link between Rhode Island and the northern Boston metro area.

KA2DEW and K1TR converted the node to two port 145.07/445.6 to tie into Mt. Washington, MBOS, CENTMA and CENTNH in the fall of 88. This was done after CENTMA was put on by KA2DEW and KA1OXQ as a three port, thus tying NH and Boston into the Eastnet Backbone Network from Mt. Greylock (WMA) and into the ALB node in Albany. Additional radios were added to SNH as time went by to create hidden transmitter free circuits from SNH to CENTNH, BERK and KNGSTN. The CENTMA link was dropped in fall 89 after BERK node came on line. Two user ports were added on 220 and 440 to tie in the KB4N and KA1GOZ PBBSs. An additional 2 meter port was added to tie in K1EA's Dx-Cluster. Recently the 220 radio that served KA1GOZ was removed to be used as a backbone link to Rhode Island. This is still pending testing and completion. KA1GOZ now shares the 440 dedicated user port with KB4N.

The purpose of this article is to bring you all up to date on the configuration of the K1TR-SNH node as this configuration is not at all obvious to a user.

Due to a design limitation of the TNC2 it is best not to have more than 5 or 6 TNCs in a single node cluster so the SNH node cluster has been broken up into 2 pieces. The pieces are linked via a modem-disconnect-header to modem-disconnect-header wireline link with two dedicated TNCs. (See illustration) The two node clusters are designated backbone port cluster and user port cluster.

The backbone cluster has four TheNET TNCs:

- the wireline link;
- a TNC which faces KNGSTN;
- one that faces CENTNH and MTUNC;
- and one that faces CHSTR.

The user port cluster has four TheNET TNCs:

- the wireline link;
- a TNC for YCCCDX;
- a TNC for SNH;
- and a TNC for SNHUHF.

The wireline link has a 3 ports:

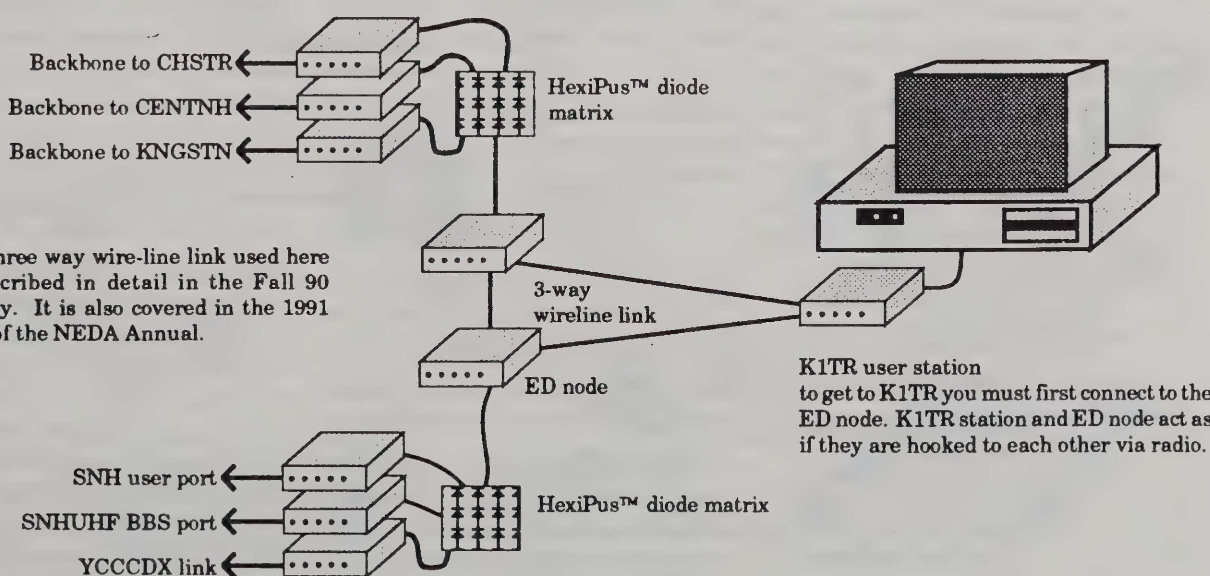
- a TNC on the backbone cluster;
- a TNC on the user port cluster;
- and a 3rd TNC which is running PacComm user station software and is K1TR's home station.

The total node has 9 TNCs and 6 radios.

The communications between the user port TNCs and the wireline link TNC is at 9600 baud. The user port TNCs are all MFJ 1270B. The Backbone TNCs and two of the three wireline link TNCs are PacComm Tiny 2s. The ED node and the K1TR user station are original TAPR TNC2s. The baud rate between the backbone TNCs is 19.2Kbaud and the baud rate between the three wireline link TNCs is 9600 baud.

Ed's user station consists of a Hewlett Packard smart terminal. Ed has a PC compatible lap top which he uses when the HP terminal isn't enough for the task.

—KA2DEW & K1TR



The three way wire-line link used here was described in detail in the Fall 90 Quarterly. It is also covered in the 1991 version of the NEDA Annual.

Bangor Pa nodes

Copied from Quarterly #2.1

As reported in a previous quarterly KA3ODJ has been busy building a multiport TheNET (etc.) node in Bangor Pa in eastern Pennsylvania. The node is entirely sponsored by Andy Horn, KA3ODJ and is located between Bangor and Stroudsburg on Kittatinny Ridge at an elevation of 1600' above mean sea level.

The system is a hybrid of technologies and is designed to be a gateway between different level3-box technologies. The current system includes 4 TheNET TNCs and a KAnode port.

The NEDA side of the system contains four PacComm Tiny 2 TNCs running TheNET. Three of the TNCs are connected to radios. The fourth is connected to port A of a NX2P Radio Multiplexer. Port B of the multiplexer is connected to a Kantronics KPC-4 dual port TNC. The KPC-4 serves as a dual port KAnode and is also hooked to a radio on 145.03. This allows TheNET users to connect to a KAnode, bridging the two technologies without an additional radio link.

Example operations:

If one wanted to connect to 145.03 from BANGR2 (2meter user LAN port) you could connect to BANGR2, then connect to KABANG (gateway TNC). KABANG allows access to any other (non-TheNET) network technologies across the radio multiplexer. If you were to do a nodes list you'd only see the TheNET nodes. The KTR node does not show up because it is not a TheNET node. If you did an I(nfo command at KABANG you'd get a message that explains that you can connect to KTR. If you do a connect to KTR you should get the connected to message from the KPC-4. A L(ist nodes can be done from here (type L). This shows what KTR can hear. All of the nodes that show on this list can be connected to using the X command (cross connect). This is because you came in on one port on the KAnode and you are leaving on the other port. If you want to connect to a user on 145.03 you'll have to use the X command for that as well. If I wanted to connect to LCNJ (Liberty Corners NJ) I would have to type X LCNJ. This will do an internal connect through the KPC-4 and then connect you over 145.03 to LCNJ.

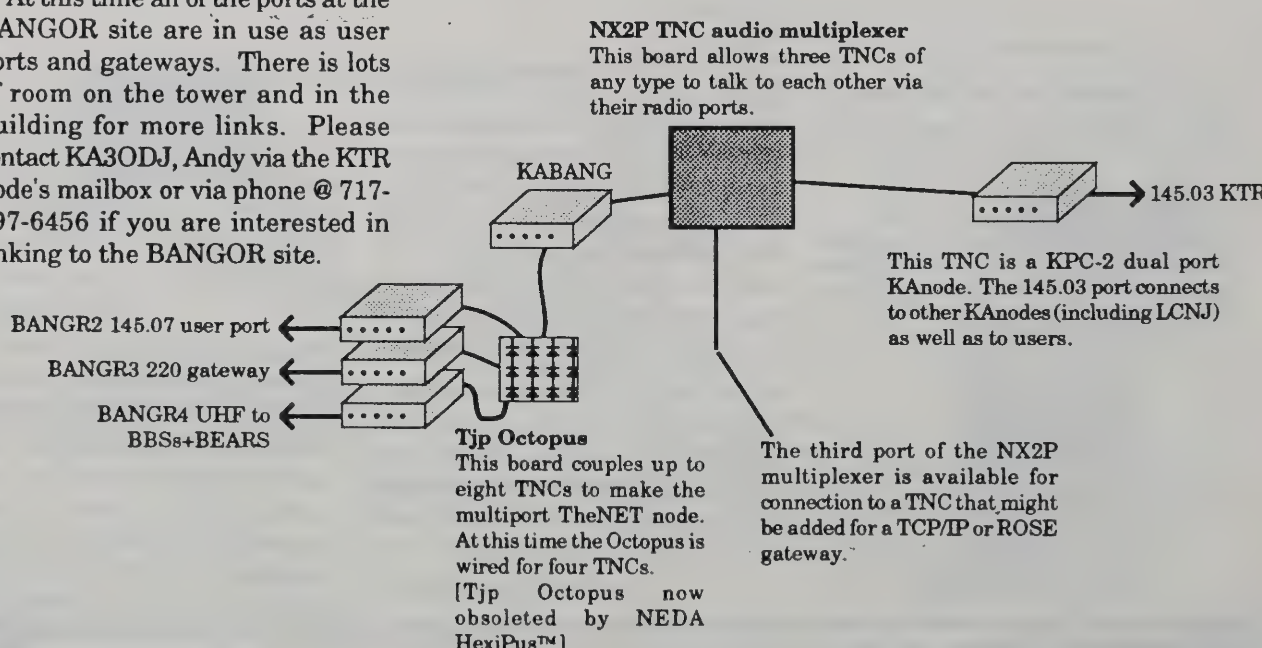
Another example: If you wanted to connect from LCNJ to BANGR3 you would first tell LCNJ to connect to KTR, then use X KABANG to cross connect to the other port on the KPC-4 and over the multiplexer. After you get the #Link Made response from the KAnode you can tell KABANG to connect to BANGR3 by typing C BANGR3.

—Andy, KA3ODJ

The node identifiers are:

TheNET	**BANGR2**	145.70	omni @ 10 watts
TheNET	**BANGR3**	223.48	beam @ 25 watts
TheNET	**BANGR4**	441.00	beam @ 4 watts
TheNET	**KABANG**	link to NX2P Radio Multiplexer	
KAnode	**KTR** (port 1)	145.03	omni @ 50 watts
KAnode	**KTR** (port 2)	link to NX2P Radio Multiplexer	

At this time all of the ports at the BANGOR site are in use as user ports and gateways. There is lots of room on the tower and in the building for more links. Please contact KA3ODJ, Andy via the KTR node's mailbox or via phone @ 717-897-6456 if you are interested in linking to the BANGOR site.



The HexiPus™ Story

Copied from NEDA Quarterly #2.1

By the summer of 1988 KA2DEW and WA2WNI had been conversing regularly via packet for about 2 years. KA2DEW lived in a rented house with John Painter in Nashua NH. WA2WNI lived with his wife and two kids in Valatie NY near Albany. The path that they used was often digipeating through two very well placed digipeaters over the distance of 150 or so miles. Other times they would maneuver their way through the 221.11 backbone and still at other times they would rely on the PBBS systems to deliver their mail overnight. Both WA2WNI and DEW had W0RLI PBBSs of their own and both ran those PBBSs with a separate 2m user channel and 440 or 220 'backbone' channel for forwarding and whatnot.

The reason that the summer of 1988 is important is that it was around that time that the level of frustration from the difficulties of trafficing mail and from trying real time packet from Albany to Nashua reached a threshold where the two decided to do something about it.

John Painter is important to this story because as he was sharing a house with DEW he had observed these goings on. John, or Tjp (THE John painter) as he likes to be called, is a technical person. He makes his living consulting to various companies that need custom VAX/VMS software to do tiny little nefarious tasks like graphics or dual ported diskdrive support etc.. John had observed Tadd, KA2DEW, on the phone with Dana, WA2WNI, for all hours of the night trying to find packet routes that work.

Now, Tadd and Dana were avid followers of the goings on in the New York metro area and had seen the application of TheNET in multiport nodes under the auspices of the Eastnet Backbone Network or Eastnet Mafia. Doug, WB2KMY (Kiss My Yagi) had taken Tadd and Dana under his wing to educate them as to technical solutions using the multiport TheNET design. Tadd and Dana learned well. One thing they learned quickly is that building those icky diode matrices is a pain in the derriere.

Tjp was in on one of these construction projects (it was unavoidable) and decided that this was a perfect application for his Macintosh and McCAD program. Presto chango cherrio and the Octopus was born. John made 200 of the boards and he made them 8 port figuring that if 4 ports was good, 8 would sell like hot cakes. This was all based on Tadd and Dana's prediction that the Octopus would make node manufacture easy enough that people would use them. [They were lying. They just wanted the boards hi]

Well.. 2 years later the Octopus boards are all sold out. It was time to make more. Many things have happened in the last 2 years. For one thing, NEDA was born out of the incredible growth in multiport nodes that occurred in the vicinity of Albany and Nashua [hmm...]. John has moved to Kansas City, Tadd recently to Potsdam NY. So.. the NEDA board of directors made a general statement that a replacement Octopus was required.

Rich, WB2JLR, took the project and ran with it, creating the HexiPus™. The reason that NEDA wanted a new board were several fold:

1> 8 ports was deemed electrically unsound due to loading problems when one TNC is driving 7 others. 6 ports was deemed more appropriate.

2> The Tjp Octopus didn't say anything about NEDA.

3> The Tjp Octopus was made small to keep costs low. The size is uncomfortable for some to construct.

4> John Painter (Tjp) wasn't around when the time came to make these boards. He has since been back in contact and is planning on working on other projects for NEDA.

So, NEDA now has 200 HexiPus™ boards. They say NEDA all over them. The club formed a committee headed by WA2TVE, Howie, to mail the boards. Orders will be processed by WB1DSW, Herb, who is the club treasurer and who picks up the mail at the POBox.

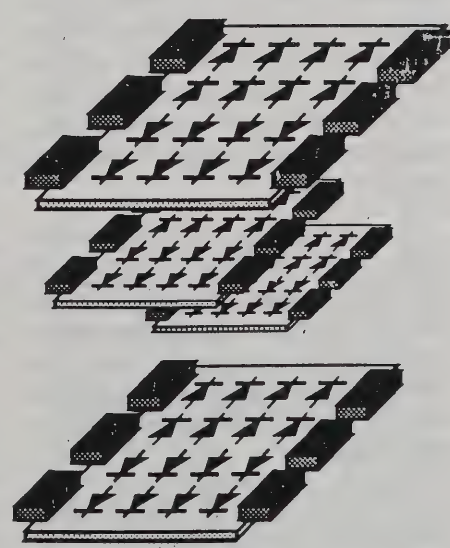
The price for the boards with diodes as a kit is $22.95 plus $4 for shipping.

The price for the boards with diodes and 9-pin Dshell connectors is $29.95 plus $4 shipping.

The Dshell connectors are female and will require a custom cable to plug between the HexiPus™ and a Tiny 2. If you are using a MFJ 1270B or AEA PK88 (for ROSE) you can get a standard PCAT to modem cable and modify it to put in the missing pin connections.

The board may be ordered without connectors if you wish to solder directly to the HexiPus™. An alternative is to put PC mount male connectors on the bottom of the board. Then you can use standard PC monitor extension cables. If you find something novel please let NEDA know.

—NEDA



Active Coupler for Mating A G8BPQ PC to a HexiPus™ TheNET/ Diode Matrix

Copied from Quarterly #2.2

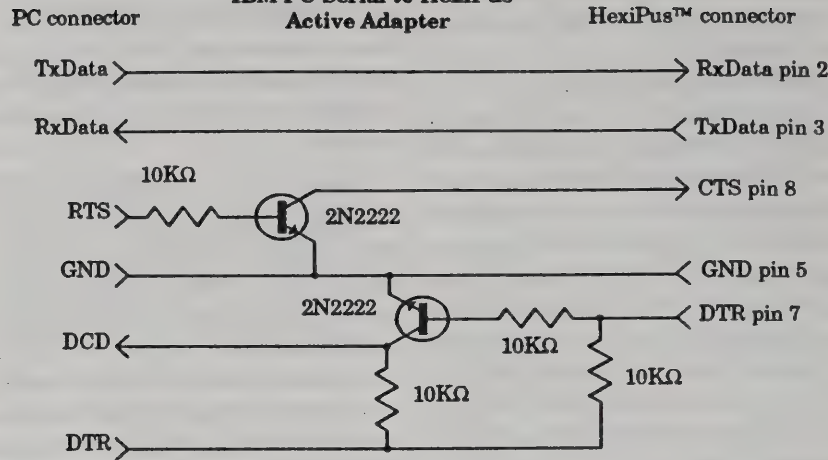
Quite a few of the switching systems in use today on the NEDA network feature not only a multiport node but also a PC-based BBS or mailbox. The node setup on most of these systems involves the use of a diode matrix to allow all the TNCs to transmit and receive data harmoniously with one another. And, most of these diode matrices are in the form of a small circuit card (like NEDA's HexiPus™) which holds the various passive components. Because these sites are often in key locations, the SYSOP can, and often does, opt to incorporate a BBS or mailbox to compliment his/her setup.

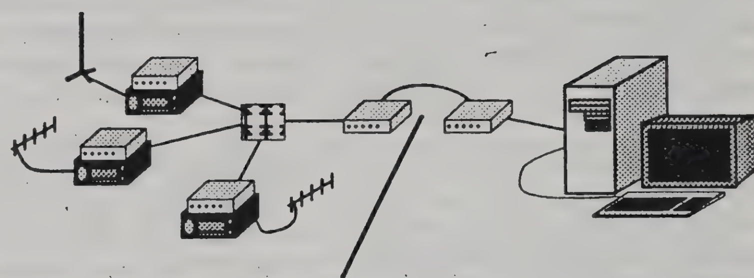
These BBS/mailbox incorporations also include in background implementations of John Wiseman's outstanding network software packet (hitherto referred to as "BPQ") to allow for multiple connects into and out of the mailbox software. BPQ's original intent was to provide an easy way for hams to create packet data switches using their already existing IBM or compatible computers. A serious drawback exists because should the computer crash for whatever reason, the switch code itself dies until the computer is reset. To be effective for a large multiport node setup, BPQ often requires a very fast (and very expensive) computer to handle the many streams of data.

A very real advantage of using the diode matrix card in conjunction with the BBS/mailbox combination is that the matrix never "dies" allowing the node to stay active even when the PC's hardware or operating system fails or is taken off the air for whatever reason.

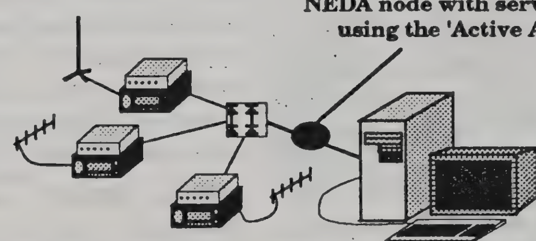
BPQ code though was thought impossible to add direct to this diode matrix. The signals just weren't compatible. To get around this and

IBM PC Serial to HexiPus™ Active Adapter





NEDA node with server connected using a wireline link.



NEDA node with server connected using the 'Active Adapter'

the more important limitation mentioned above, NEDA devised a scheme whereby placing two TNCs "front-to-front" would allow for the node to exist on one side and the BBS/Mailbox to exist independently on the other. By facing these TNC's HDLC (radio) ports toward each other, the RS-232 ports became usable - one looking toward the matrix and one toward the BPQ code and the associated BBS/mailbox.

As this idea developed, we progressed from simply cross wiring the Tx/Rx lines off of the DIN connectors to picking the digital Tx/Rx signals off of the modem disconnect headers inside the TNCs. This allowed us to set the baud rate of that link up to 19200 baud.

Two advantages were apparent. First, the computer could crash all it wanted but the node stayed a viable, active part of the network. Second, the computer now was not burdened with the awesome task of switching data between its serial ports plus doing the work on the BBS. However, this involved the use of two TNCs - a couple of hundred bucks just to do this? Boy, was this expensive! There must be a better way.

Enter yours truly. I had a spare older XT and a temporary allocation of TNCs connected to the matrix to play with and besides, I had an afternoon to kill.

I reasoned that since BPQ code

An Improvement to the Wink-N-Blink Mod

Copied from NEDA Quarterly #2.1

In the last Quarterly KA2DEW provided a simple TNC conversion that modified the STA and CON LEDs to show RX and TX data being passed over the RS-232 port. This was part of his 'Wireline Linking' article. As you may know a TNC running TheNET code does not normally use the STA and CON. The WNBM (Wink-N-Blink Mod) can be very useful as a diagnostic tool to show matrix activity such as which port is sourcing matrix data and when.

I did find it aesthetically more pleasing however to remove the right hand 3 LEDs in my Tiny 2s and rearrange them so the LED color order from left to right across the front of the TNC was Green Red Green Red and the power LED became the remaining Yellow. The 1st pair of Green/Red show RF port Rx/Tx data and the 2nd pair then show the Rx/Tx data appearing on the RS-232 port. When the TNCs from your node are physically stacked on top of one another it creates and easy to visualize pattern so you can quickly familiarize yourself with what normal data throughput should look like. (Not to mention the fantastic light show that will entertain visitors to your node setup when the node matrix is extremely busy!)

Several useful things cropped up almost immediately after I had put the WNBM in place. First was that I noticed that one port was sending many Tx bursts without responses from anything. This could not possibly be caused by **that** many matrix collisions in a row! Indeed it wasn't as I quickly found that there was an open connection across the matrix between two TNCs. The receiving TNC wasn't getting the data at all so the sending TNC kept trying until a the alternate default path took it a different way around the matrix.

The second item of interest was to actually observe 3 TNCs passing data that should only have gone through 2 TNCs. Alas there was a routing glitch caused by a locked value. Data in one direction was going through TNCs A and B while the return data was going through TNCs A to C to B. Of course one can observe by looking at parm and route tables all this anyway but it takes a bit of looking to see what paths are set up to do what.

One of the observations I am keeping an eye on now is the effect of circuit choke on the matrix. I have noted at times the RF ports doing all sorts of activity, and the matrix doing nothing at all. Hopefully if somebody gets around to building the MoniPus product I'll get a better view of all this. The MoniPus is a project that has shown up in the NEDA minutes recently.

—Dana Jonas, WA2WNI

---

was basically simulating a TNC-like device, why couldn't it be directly connected. I quickly drummed up a working version of the BPQ code and configured one TNC (with TheNET 1.16) as a node via the matrix card. As a start, I just tied the various lines off of a vacant port of the matrix to match the RS-232 lines on the PC's serial port. Needless to say it didn't work. So I started taking things off and rearranging them. When I took the RX and TX data lines and reversed them, it started working! After a brief period of testing, I published my findings to NEDA @ W1NY and got some enthusiastic bravos.

Rich Place, WB2JLR made one important addition however which I didn't account for. The DCD circuit was doomed to fail in that my little "experiment" didn't incorporate more than one TNC. Rich was certain that placing the PC on the matrix with more than one TNC would cause problems without correctly using the DCD/CTS collision detection scheme. He drew up a simple way to use the DCD circuitry between the PC and the matrix using two 2N2222 transistors and some pull-up resistors. This circuit inverts the states of the DCD and CTS lines from the PC, making them work with the DCD and CTS from the TNC 2s.

Since implementing this, I have regained two shiny new TNCs for use elsewhere, reduced the load on my 12V DC power supply by 2 TNCs and eliminated a source of traffic delay between my BBS and the rest of the network. (Someone recently pointed out that they had noticed an overall improvement in the speediness of the return data). As an extra added bonus, by setting the BPQ code up with certain parameters turned on, I get to see all the L3 and L4 activity between the BBS and it's users as well as all internode traffic and TheNET activity between the TNCs on the matrix, something I was unable to do previously. One down side of this method is that I lose my "Mail For:" beacon (who cares!).

My thanks go to Rich WB2JLR and Mike N1BEE for their help and encouragement. Special thanks to Tadd KA2DEW and Rob KC3BQ for the original thoughts and work on getting this system off the ground.

—Herb Salls, WB1DSW

Kantronics D4-10 UHF Radio

Copied from Quarterly #2.4

These are some notes based on our experiences (over the last 10 days) of making the new Kantronics D4-10 radios work at 19.2 KB. Our group is using these radios to build a metropolitan area network that includes a full duplex UHF digital repeater, a G8BPQ switch, and high-speed links to other areas.

The Hardware

The Kantronics D4-10 radio (not to be confused with the 2M DVR2-2) is a UHF radio designed for data transmission. Kantronics has optimized the D4 to move data at 19.2 kilobaud within a 100KHz channel with a 60KHz receiver bandwidth. It's crystal controlled on two channels nominally in the 430 MHz range and is rated at 10W output, although my Bird says more like 15. It has a <much> better receiver than the 2M DataRadio.

The interesting feature of the radio is that it has a TTL level I/O port designed for direct FSK. TXD will modulate +/- 10KHz around the center frequency, and RXD is derived from a data slicer. The squelch circuit is very fast (~2ms) and is available as DCD on the connector. And not to worry — the TXD line is shaped, so the FSK isn't based on square waves. The bandwidth is within FCC limits (100KHz) for the 70cm band.

Our Approach

Since the radio is designed with digital levels in mind, my first testing with two of the beta models last March focussed on the simple approach — using an 8530 SCC chip to generate HDLC frames and shoving those frames directly into the D4 TTL port. To my surprise, it worked!

Since then, we've decided to base our network, at least for now, on that approach. If and when modems arrive that can do a better job, we'll probably use them, but for now the savings of $100 per radio by not buying 19.2K modems outweighs the relatively small advantages the modems offer (mainly in more reliable DCD, but even that's open to question).

Using the Ottawa PI Card

Our first experiments used the Ottawa packet group's PI card (a DMA driven, 8530 plug-in card for the PC bus). Interfacing them to the D4 is a snap — just wire up a five conductor cable between the two, set up NOS, and you're in business.

Interfacing the Data Engine

However, the PI card only works in PCs, and (at present) only works with the KA9Q NOS software. We wanted to have an alternative packet generator available, so we focused on interfacing the DataEngine to the D4, sans modem. That also proved easy to do.

Kantronics makes a small jumper board (for about $25) that's designed to allow the DataEngine to work with an external modem. Just get one of those, jumper it as a type "A" modem, and add a CMOS chip to divide the RXClock signal by 32 to feed back as TXClock.

More specifically, we used a CD4020 with the clock connected to pin 5 of the internal modem header and the divided output connected to pin 6. 12 volts is available on the jumper board; we used a 1K resistor and 5.1 volt zener diode to power the 4020 chip with the necessary TTL level. The chip can be mounted "dead bug" style on the jumper board; the whole thing makes a very nice package.

Software Speed Selection

With either of these approaches, the actual data rate on the radio link is totally software-driven. It's just a matter of what speed you program the baud rate generators to. We've moved packets at every supported rate from 110 baud to 28.8kb (28.8 doesn't work very well, but it does work), simply by using the appropriate "attach" command with NOS, or "modem" command with the DataEngine.

Results

First, these radios are as fast as Kantronics says they are. The PI card driver allows TXDelay to be set in 1ms increments, and we've found that a TXD of 4ms works. We're using 5ms to provide a bit of margin. Remember, this is <milliseconds>, not the (milliseconds times ten) that most TXD values represent.

Our initial testing shows that very respectable throughputs are easy to achieve, at least across the room. Using NOS on 286 or better machines, and a RAM disk to avoid mechanical slowdowns, we've consistently seen FTP file transfers of binary files go at 1600 or more characters per second between two PI cards. Note, though that this is on a totally clear channel, with all parameters set wide open. In the real world, neighborliness will require backing things off a bit.

We do see some packets that don't get acknowledged; the resultant retries and backoff can slow things down a bit. We're investigating the problem, but at the moment don't have any clues.

We only began testing the combination of a PI card station talking to a DataEngine station last night. The throughput there has been more like 650-700 characters per second. We're not sure why this great a difference exists. Possibly the problem is that the DataEngine-to-host link on the serial port is running at the same speed as the radio link, that the computer just can't keep up with 19.2 serial data (we're not using 16550s, so even though the machine is a 386, this is quite possible), or that the asy code in NOS is be less efficient than the PI driver. We're going to continue looking at this.

Digital Repeater

We're turning two of the D4s into a digital repeater. Our input frequency is in the 420 MHz range, with output on 430 (a 10MHz split). The interface is actually very easy but it took a <lot> of trial and error to get things working.

The problem in a nutshell is that although the digital port is advertised as "TTL", it really isn't. The PTT line is fairly standard — to key the radio, bring the line to ground and sink about 5ma.

However, the DCD, TXD, and RXD lines are all tied to op amp stages set up as comparators. Although they are biased to switch with TTL levels, we found that using 13.8 volts is much more reliable.

Also, it's not obvious from the documentation but the FSK keying circuit actually has <three> states, not two. Grounding TXD shifts 10KHz down, pulling it high shifts 10KHz up, and something between will put out the nominal frequency. This cost us a lot of time — our first interface <seemed> to be modulating the radio, and we could hear the data on the receiver's speaker, but there was no RXD. It turned out we were shifting between -10 and center — enough deviation to make noise, but not enough to trigger the data slicer.

Anyway, the answer was simple once we figured it out. We used a CD4049 hex inverter chip. Two cascaded sections provide PTT from the DCD input. Two more sections interface RXD to TXD. The chip is powered from the same 13.8 volt supply as the radios.

The critical thing it took a while to figure out is that RXD <must> be tied high at the input to the inverter. Not doing this is what caused our indeterminate keying state. 22K between Vcc and RXD worked fine for us. DCD would probably also benefit from a pull-up resistor, but seems to work OK without one.

Of course, you'll need extra circuitry for control and time-out timers. We're also looking at ways to come up with a more reliable keying scheme; if the repeater is brought up by a rogue carrier, that will shut the whole network down. A circuit that detects a packet's opening flags and trips a short timer (maybe 1 second) AND'ed with the squelch-derived DCD is probably the simplest answer.

The repeater turn-around is pretty quick. We've been able to reliably send packets through it with a TXDelay of 10ms. Obviously, a hang-timer won't work in a system based on carrier-derived squelch, so the repeater output is indistinguishable from any other packet station.

Repeater identification will be handled by the G8BPQ node that will be interfaced with the repeater.

—John Ackermann, AG9V, and the Miami Valley FM Association, Dayton, Ohio". Republication and distribution is no problem so long as credit is given.

New Deal Available on UHF HTs

Copied from Quarterly #2.2

About a year or so ago some NEDA member may recall how we were frantically buying up Wilson MH400 UHF HT's that we then put into links in a number of places. WA1TPP did an article for the Quarterly that showed us how to speed up and utilize the rig for packet as efficiently as possible and thus a number of links and special ports were put on the air as a result of these inexpensive little 2 watt hand held rigs. (The rigs only cost us about $80 each!)

Well, it appears that the original vendor has struck yet another bargain with the folks at the Wilson/ Regency warehouse and managed to make another incredible bulk buy out. The deal this time is for UHF Regency MCPU-404 handhelds. These rigs are new closeouts that are 4 channel, 4 watt crystal controlled handie talkies. While they don't come with antennas or batteries, the vendor is selling them to us for the incredible price of $49.95 each plus shipping. If you should choose to use the rig as a portable and not cannibalize it for a packet link the vendor will sell you a drop in charger for $29.95 and batteries for $20 each. The batteries, by the way, are the same ones used by the Yaesu FT 203 and 209 series radios, known commercially as a BP-4. The crystals it takes are HC-18u size with wire leads and the unit also has a factory installed jack that will take a plug in CTCSS board.

Please contact the vendor directly as we will not have sufficient time to put together a NEDA bulk buy like in the past. He was kind enough to let us in on the lower pricing because of being on his preferred buyers list from the previous 30 or 40 odd units we bought before. Please move quickly on this as the vendor might do something like raise the price or sell the whole lot to someone. The address:

Torg's Electronics
9280 W. 360N
Shipshewana, IN 46565
Phone: (219) 768-4406

The proprietor is Mr. Torgeson so should you give him a call to get in on this, make sure to pass on the regards and best wishes for all the NEDA members already taking advantage of the bargains he has provided us in the past. Who knows? With a little effort maybe we can get Mr. Torgeson to get his ***own ham license***!?!

Oh, one more thing. Manuals for the unit should be readily available from *Regency Electronics*, or Torg's can most likely provide you with a copy for a nominal extra cost.

—Dana Jonas, WA2WNI

Continued on page 4

Taking Your First Steps with Packet Radio

Copied from Quarterly #2.1

You've heard a lot of interesting talk about packet radio or you've seen packet demonstrated at a friend's QTH, so now you've decided to buy a TNC and get on the air yourself. What kind of TNC should you buy? There are so many different types on the market now it can be a difficult decision to choose just the right one. First you have to decide whether or not you want just packet, or whether you also want RTTY, AMTOR, baudot, facsimile, and CW too. You can buy a "packet only" TNC for less that $150 or you can spend more than double that amount for an "all mode" unit. Shop around, ask others who are already on packet for their opinions, and then choose the one you feel is best for you. If you plan to use the TNC on the low bands, you'll need to make sure that the one you buy is capable of tuning HF. Many units are made for VHF FM use only.

When you buy your TNC you'll find that cables are supplied with it for connecting the unit to the radio, but you'll have to buy the appropriate mic and speaker jack connectors for the radio you're going to use. You also have to furnish the cable that connects the TNC to your computer or terminal. In most cases, the standard RS-232 port is used between the TNC and computer, however this varies with the type of computer and TNC used. The operating manuals supplied with the TNCs have a good write up on the various computers and the cabling needed. I would advise that you read the introductions and set up procedures for your particular TNC very carefully. Most companies have supplied excellent manuals, and you usually can figure out everything you need to do to set up your TNC from the information supplied in the manual.

Once you have everything wired and connected together, turn on the computer, load a terminal program (anything used for a phone modem will work well for packet) and get into receive mode. Now turn on the radio and make sure the volume is turned up about a quarter turn (about the "10 o'clock" position) and make sure the squelch is set. It should be at the point where the background noise disappears, just as it would be set for a voice QSO. Next, turn on the TNC. You should receive a "greeting" or sign on message on your computer screen showing the manufacturer's name, software version etc. If you see a bunch of gibberish it means that the data rate of the TNC and computer are not the same. This data rate is better known as the baud rate. The baud rate of the TNC has to match the baud rate used by your computer terminal program. This is easy to adjust. You can change the terminal program to match the TNC, or vice versa. Check your TNC manual for the procedure as it varies from TNC to TNC. If you don't see a "greeting" or gibberish, check your cables and connections. Make sure that you have everything connected properly, that the right wires are on the right pins, and that everything is secured tightly.

Now you need to know about the three levels of communicating you can to from your keyboard. First, you can communicate with your computer for setting up the terminal program; second, you can communicate with the TNC; and third you can communicate with the radio. It's very important that you know which level you're in when working packet. I can't help you much with the computer level, since that varies with the manufacturer, model and the terminal program you're using, but once you get the terminal program ready to receive data, you're ready to talk to the TNC.

First, do a "control C" (simultaneously press the CTRL and the letter C keys); this puts the TNC in COMMAND mode, the level where you communicate directly with the TNC from the keyboard. You should see "cmd:" on your screen. Enter "MYCALL xxxxxx" replacing the x's with your callsign, such as "MYCALL WB9LOZ" followed by a carriage return (CR). All commands are followed by a (CR). This sets into the TNC memory the call that you're going to use on the air. Now if you type "MYCALL" (CR), it should respond with your call. If it does, you've proven that the computer to TNC linkup is working fine. If you do not see anything on the screen when you type, blindly enter the following: ECHO ON (CR). If you see two of everything that you type, such as "MMYYCCAALLLL", enter ECHO OFF (CR).

You're now ready to go on the air! Tune the receiver to any odd numbered frequency between 144.91 and 145.09 that has some activity on it and set the rig up for simplex operation. Enter "MONITOR-ON" (CR), then watch the screen. You should soon be seeing the packets that are being sent over the air by other stations. If you don't see anything for a minute or two, try tuning to another frequency. Watch for callsigns and jot down a few of the ones you see, including any number at the end of the calls.

In packet, you can have up to 16 different stations on the air at the same time using the same callsign. That's where the numbers in the callsign come into play. The calls W6PW, W6PW-1, W6PW-2, W6PW-3, W6PW-4 and W6PW-5 are all individual stations operating under the same station license. A callsign without a number is the same as -0. The numbers are used to differentiate between the various stations and are called SSIDs (secondary station id).

Making Your First Contact

Now it's time to make a packet QSO. As you monitor the frequency, watch for familiar callsigns, someone completing a QSO with another station or someone sending a CQ. It might be a good idea to make arrangements with another packet station that's near by to get on the air and help you with your first QSO. There are packet nodes, digipeaters, personal mailboxes and bulletin board systems that are on the various frequencies, and often it's not possible for a new packet operator to distinguish between one of these and a "regular" packet station. You can work these stations later on, but for your first QSO it's best to work a station with someone operating from the keyboard at the other end.

Choose the callsign that you want to try contacting and enter "C xxxxxx", replacing the x's with the callsign you've chosen. The C means CONNECT, "C WB9LOZ" means connect to WB9LOZ. You should soon see *** CONNECTED TO (callsign): on the screen. You have now entered the third level of communications, called CONVERSE mode, and this is where you communicate from the keyboard to the radio. Anything you type on the keyboard will be displayed on your screen and every time you hit a (CR) the information will be transmitted over the air as a packet. Likewise, anything

transmitted by the other station will be received by your TNC and also displayed on your screen.

When you have completed your QSO, hit a CONTROL-C. This puts you back into COMMAND mode where you talk to the TNC again. Enter a "D" (CR). This will disconnect you from the other station and you see "DISCONNECTED" on the screen.

You're on the way now to lots of packet fun and adventure! If you are still having problems at this point, contact a friend that has some experience on packet and ask for help. The initial set up of the computer, TNC and radio is probably the biggest stumbling block in packet. Any experienced packeteer will be happy to help you get through this process to get you on the air.

Using Digipeaters and Nodes

Digipeater is a term we use to describe a packet radio digital repeater. Unlike the FM voice repeaters, most digipeaters operate on simplex and do not receive and transmit simultaneously. The receive the digital information, temporarily store it and then turn around and retransmit it.

Your TNC will allow you to enter up to eight digipeaters in your connect sequence, but using more than 3 usually means long waits, lots of repeated packets, and frequent disconnects, due to noise and other signals encountered on the frequency.

When entering the list of digipeaters in your connect sequence, you must make sure that you enter them in the exact order that your signal will use them. You must separate the calls by commas, without any spaces, and the EXACT callsigns must be used, including the SSID, if any. This means you need to know what digipeaters are out there before you randomly try to connect. Turn MONITOR ON and watch for the paths that other stations are using.

Here are some examples of proper connect sequences:
C W6PW-3 V W6PW-5
C N6ZYX v WA6FSP-1,WB6LPZ-1
C W6ABY-4 V W1NY,AB2L,K1TR

The v means via. In the first example the sequence shown means: Connect to W6PW-3 via W6PW-5.

Something to remember when using digipeaters is the difference between making a connection and sending information packets (those with text in them). If the path isn't all that good, you might be able to get a connect request through, but will have a difficult time with packets after that. The connect request is short so it has much less of a chance of being destroyed by noise or collisions than a packet containing information. Keeping information packets short can help keep retries down when the path is less than ideal

Packet Node Network

NET/ROM, TheNET, G8BPQ packet switch and KA-Node are names that refer to a device called a packet node, another means of connecting to other packet stations. Each node has several functions available, but for now we'll cover the basics so that you can try them out. The difference you should note here is that you connect to a node, rather than using it in a connect path as you do with a digipeater.

Here's how you use the packet node network: First you connect to the closest node to you on the frequency that is clearest. You connect to a node the same way as you connect to any other packet station. When you connect to a node, your TNC automatically switches to converse mode, so anything you now type is sent to the node as a packet, and the node acknowledges each packet back to your TNC. For the remainder of your connection your TNC works only with this one node.

Once you're connected to the node enter "NODES" and you'll receive a list of other nodes available to you on the network. This list will give both an alias ID and callsign for each node. The alias ID often gives you a hit as to where the node is located, but not always. Descriptive node listings giving the alias, callsign, location, frequency and other information on the nodes are available on most packet bulletin board systems [ed: See maps in this issue].

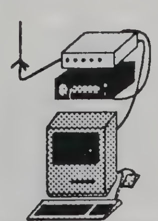
Now that you have a list of nodes what do you do with it? Let's say you want to have a QSO with another station. You first must determine which node from this list you received is closest to the station you want to work. For demonstration purposes, we'll connect to N6ZYX, located near the W6AMT-3 node.

Once you know the call of the node you want to use, you connect to it *while still connected to your local node*. You use the standard protocol: C W6AMT-3. Your TNC will send this as an information packet to your local node, and your local node will acknowledge it. Your TNC is happy because the packet has been sent and acknowledged. The network will then go to work for you and find the best path between your local node and the one you're trying to reach. You'll then see one of two responses: "Connected to W6AMT-3" or "Failure with W6AMT-3". If it can't connect for some reason, try again later. It could be that W6AMT-3 is temporarily off the air or the path has decayed and is no longer available. We're going to be positive here and say we received the first option.

Now that you've connected to W6AMT-3, enter "C N6XYZ". Again, your TNC will send this as an information packet to your local node and the node will acknowledge it and send it down the path to W6AMT-3. W6AMT-3 will then attempt to connect to N6XYZ". If you get connected, you hold your QSO just as you normally would, but there's one BIG difference — your TNC is receiving acknowledgments from your local node, and N6XYZ is receiving acknowledgment from W6AMT-3. The acknowledgments do not have to travel the entire distance between the two end stations and the long path is eliminated for both TNCs. Because of this, retries are greatly reduced, and your packets get through much faster. When you're finished with the QSO, you disconnect in the normal manner — go to Command Mode using Control C and enter "D". The entire path will then disconnect automatically for you.

–WB9LOZ

NCPA Education Coordinator from the NCPA Downlink, Fall 1990



Network Development in the Ottawa Area: 1991 Update

Copied from Quarterly #2.2

Barry McLarnon, VE3JF
Ottawa ARC Packet Working Group

The Local Ottawa-Area Network

The Ottawa area is served by one major node site, plus a number of subsidiary nodes. The major "hub" site, at Carleton University, is the home of the Hydra packet switch. Hydra actually consists of two separate systems. The switch itself ("hydra-gw"), which is a PC AT running KA9Q NOS, currently has four ports. Two of these are 9600bps serial interfaces into the 2m NET/ROM nodes OTTAWA (145.07 LAN) and CAPITL (145.01). The third port uses an Ottawa PI board to interface into the 56Kbps LAN, which is served by a full duplex cross band repeater (220.55 in, 433.44 out) at the same site. Finally, there is an ethernet port which is connected to the Carleton campus ethernet. Also on the ethernet is the second part of Hydra, a Sun-2 workstation, which is a Unix system with a large amount of disk storage. This system will be the platform for developing various services for the amateur packet community. In addition to various servers such as on-line callbook lookup, the possibilities include a gateway into the Internet itself.

The 56Kbps full-duplex network is, as far as we know, the only one of its kind in the world. It began as a high-speed LAN for the "power users", but it has evolved to a combination LAN/local backbone network. This goes against the conventional wisdom of keeping LANs and backbones separate, but it is successful because there are no hidden transmitters, and the capacity is more than sufficient to handle both functions. When the 56Kbps network begins to get congested, our plan is to "twin" the cross band repeater with a second one, using additional 100KHz channels in the same bands. It is remarkably simple to add a second repeater in this way, since the antennas and RF gear can be shared between the two repeaters, with power combining/splitting done at the 28-30Mhz IF of the 56Kbps modem.

The 56Kbps network provides the link from the Hydra switch to the three Ottawa area BBS stations (VE3JF, VE3NAV, and VE3KYT), to users on two additional LAN frequencies (144.91 and 145.03), and to a conference node. This network offers an easy means of "spreading out" the 1200bps 2m traffic so that low-speed users can continue to get adequate access to the network. A user on the 56Kbps network can attach a 2m port to his station and open up a network access port on a new frequency for users in his area. This is what you might call a "cellular LAN" approach. In a traditional LAN with a wide-coverage node, modelled after voice repeaters, you have too many users, too many hidden transmitters, and therefore many collisions and a terrible throughput. In a cellular LAN with more limited coverage, you have fewer users, and since they are located in a smaller area, less chance that they are hidden from each other. An additional benefit comes from the fact that the cellular nodes are located at home stations, and therefore are easier to maintain.

The basic model for network development in the Ottawa area therefore is:

(1) A central switch with expandable capabilities, and offering access to various services for network users.

(2) One or more high-speed full-duplex repeaters which link the switch to other area nodes, as well as to some individual users.

(3) A number of low-speed limited-coverage network access nodes, on different frequencies, with the frequencies reused as appropriate. Each frequency has one, and only one, node for a given "cell", so that there is no node-to-node traffic on these frequencies.

The main departure from this model at the present time is on 145.07, where the OTTSAT node, which serves as the access point to the Calgary-Ottawa "wormhole', resides in addition to the OTTAWA node. The OTTSAT node is expected to be removed from this frequency sometime in 1991, and it will either be added to the 56Kbps LAN or provided with a dedicated point-to-point link from hydra.

Trunk Links to Other Areas

Improving our links to other areas is a high priority for the Packet Working Group in 1991. Other than the Calgary link, all out-of-town linking remains dependent on the grossly overloaded 145.01 network. One reason we have been slow to upgrade these links (other than our preoccupation with the local high-speed network and switch development) was the possibility of obtaining additional satellite links, from the OTTSAT gateway to Montreal, Toronto, and possibly other points. This has failed to materialize, and although chances are still good that something may happen, it is clear that we can no longer afford to wait. Furthermore, we should not let the possibility of using commercial satellite channels for some of our links divert us from the goal of building an autonomous fully-connected amateur network.

We are anxious to work with neighboring groups to install backbone links for trunking packet traffic between Ottawa and the surrounding areas. We do recognize that any collision-free backbone link, even if only 1200bps, would be a vast improvement over using 145.01 and would do a reasonable job of handling the current volume of BBS mail. However, we also feel that we should aim for much higher performance. Not only will the amount of mail and bulletin traffic increase quickly as the link capabilities improve, but users will require more throughput for applications such as file transfers and logging into remote servers. We feel that 9600bps should be regarded as a minimum standard for trunk

linking two major network nodes, and our preference would be to have 56Kbps on these links before long. We would therefore urge that network planners who feel that it is not feasible to go the higher speed immediately at least give serious consideration to providing an easy upgrade path.

Providing an upgrade path involves two key issues:

(1) Selecting a band (or bands) in which at least 100khz bandwidth channels are available. This means putting the link on a frequency above the 2 meter band!

(2) Designing sufficient margin into the link such that it can be upgraded to 56Kbps without changing the antennas and feeds.

With regard to the second point, there is a convenient rule of thumb: in order to work adequately at 56Kbps, the link will require approximately 10db more margin than is needed for 1200bps AFSK. For example, if a link works okay at 1200bps with low-gain omni antennas at each end, then replacement of the antennas with small yagis should provide sufficient margin for upgrading to 56Kbps (assuming the same power levels).

There are a number of reasons that the Ottawa working group has a strong preference for using the WA4DSY 56Kbps modem in linking projects. After working with the modem for nearly 3 years, we have a good deal of experience with it, and a high degree of confidence in its capabilities and reliability. It offers much higher value in terms of bits per second per dollar of investment than the lower speed modems, and its higher throughput means a longer lifetime before obsolescence. It is very easy to deploy, since it is a self contained RF modem which does not have to be interfaced to standard radios - its 28Mhz IF is simply converted to VHF/UHF using a standard transverter (or separate receive and transmit converters, in the case of full duplex). And it will run full duplex with no difficulty, unlike some lower speed modems.

The use of speeds of 56Kbps or more necessitates the upgrading of nodes with more capable packet switch hardware than the TNC2. Like the Ottawa Hydra switch, a multiport node can be configured fairly inexpensively around a PC AT class machine. The TNCs can be retained to handle the low speed nodes. For major node sites with multiple 56Kbps (or higher speed) ports, a more attractive proposition is the Grace PacketTen packet switch board. The latter board can make use of a PC as a host, so again there is a clear upgrade path if a PC is used for the switch.

—VE3JF, Barry McLarnon

*[**Editor's note**: At the time that he wrote this article VE3JF was not a NEDA member and had not read the club literature. My personal belief is that the NOS switches that Barry is suggesting as node sites are indeed the way of the future. See the TCP/IP section. Grace makes a stand-alone version of the PacketTen which is capable of handling the rougher environments and will run NOS without the PC. I would also not like to discourage someone from putting up a node just because it can't be upgraded to high speed. I don't think that Barry meant that either!*

Board members of NEDA have made strong statements in regards to packet radio connections to the Internet and there are many technical implications that must be worked out in order for such a gateway to be legal.

Thank you very much for this contribution Barry and hats off to the Ottawa crew for an excellent city-wide network!]

G8BPQ Version 4.04 HexiPus™ Interface Bug

Copied from Quarterly #2.3

We had a problem interfacing G8BPQ's switch software to our HexiPus™ which everyone needs to know about. As you probably aware, we have had numerous successes in mating a BPQ equipped computer direct to a node stack. There now appeared to be a "difference" in BPQ's handling of such things in his latest version, 4.04 causing the BPQ code and the node stack to stop talking to one another. The previously released version of BPQ (version 4.03a) worked just fine. Upgrading to version 4.04 using the same config file (there doesn't appear to be any difference in BPQCFG from 4.03a to 4.04) brought everything to a halt. Reinstalling 4.03a, resurrected the complex entirely.

A letter from G8BPQ to N2JHJ dated December 12 1991 said that a fix in 4.04 that allows for telephone modem use made it necessary to have the CTS connected to the RTS on the PC side of the HexiPus™ - PC interface cable. The fix is to tie CTS to RTS at the PC's connector. This will allow use of BPQCODE version 4.04 with the HexiPus™ once again.

John, G8BPQ, sent his apologies for overlooking adding this change to the recent BPQCODE documentation.

—Herb Salls, WB1DSW

Bug! MFJ's in Node Operation

Copied from Quarterly #2.3

MFJ TNCs wired incorrectly may be causing massive slow-ups in multiport nodes. A miss-documented pin on the 25 pin connector could be causing collisions on the matrix of nodes with more than two TNCs. The Octopus, HexiPus™ documents (and MFJ man-ual) describe pin 20 as the RTS line when actually pin 4 is the RTS line. This causes some TNCs to not know that another TNC is already talking on the Matrix. The fix is to simply jumper pin 4 and 20 on all MFJ TNCs' DB25. Also do the Wink-N-Blink mod described in the Annual.

—Tadd Torborg, KA2DEW

KNOX:WB2QBQ Node

Copied from Quarterly #2.3

Bob, WB2QBQ owns and operates the KNOX node in Knox New York. The node is located in and around his house. He is currently operating seven radios and antennas for the node itself and other radios and antennas for packet, FM and HF. The system has sprung up in a very short time, from a digipeater in 1989 to a seven port node in the summer of 1991. Bob spends a great deal of his ham radio time tinkering with the radio and antenna systems as well as playing on packet. He also derives much enjoyment from having his system used by others as is evidenced by the amazing light display on the TNCs as the node is operating. Visitors to his site have remarked at how the action never seems to stop.

Servers that are almost always using the KNOX node as a through path include the K2TR DxCluster, WA2TVE BBS, WA2PVV BBS, and WA2UMX BBS.

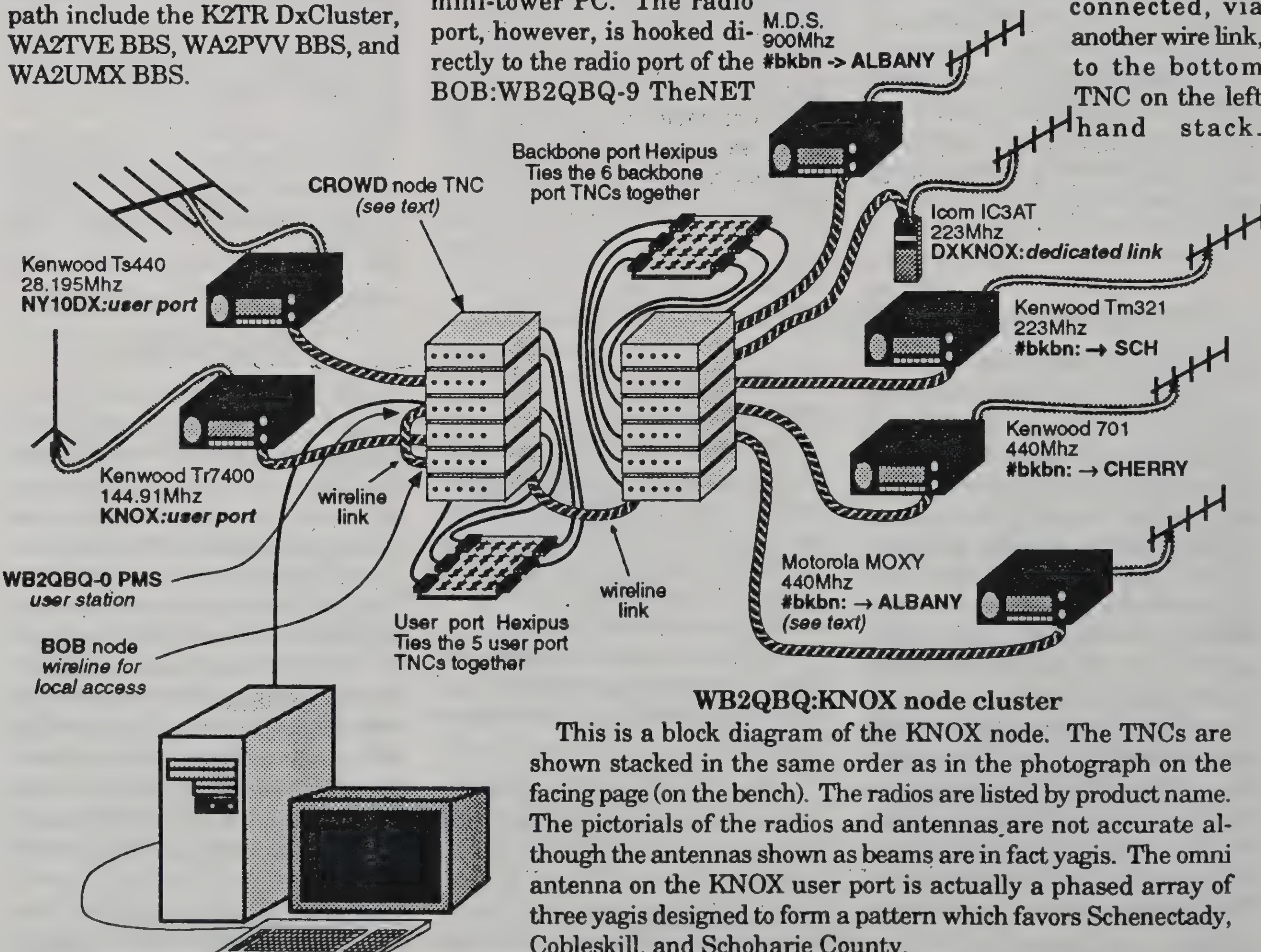
Recent additions to the node include a 900MHz 9600 baud link to Albany. The pre-existing 440MHz 1200 baud link is still in place but will be reallocated once testing on the 900MHz link is competed. Bob expects to link to another site using the already in place 440 gear.

A recent problem that Bob had is that his TS-440 HF radio has failed. As of this writing that radio is on the way to Kenwood but will hopefully be back on at or shortly after publication date.

Wireline Links

KNOX node has two separate wireline links. The first connects the home station with the network node. Bob's home station is the third TNCs down in the left stack as seen in both the photograph and the diagram. The TNC has regular PacComm PMS software and is hooked to Bob's mini-tower PC. The radio port, however, is hooked directly to the radio port of the BOB:WB2QBQ-9 TheNET TNC. Stations who connect to KNOX and do a nodes list will see the BOB node. If you connect to the BOB node and then connect to WB2QBQ you will be connected to Bob's home station and the "*** Connected to" message will appear on the PC's screen. However, there are no radios in between WB2QBQ-0 and BOB: WB2QBQ-9. The circuit is made with a few 12 inch pieces of wire connected to the modem headers of the two TNCs. That connection is described elsewhere in this issue. (See *Wireline Link*).

The other wireline link at the KNOX node is between the two separate node stacks. The right hand node stack consists only of backbone links and the DXKNOX node. The bottom TNC in the right hand stack is connected to the other five TNCs through the HexiPus™ but is not connected to a radio. Rather it is connected, via another wire link, to the bottom TNC on the left hand stack.



WB2QBQ:KNOX node cluster

This is a block diagram of the KNOX node. The TNCs are shown stacked in the same order as in the photograph on the facing page (on the bench). The radios are listed by product name. The pictorials of the radios and antennas are not accurate although the antennas shown as beams are in fact yagis. The omni antenna on the KNOX user port is actually a phased array of three yagis designed to form a pattern which favors Schenectady, Cobleskill, and Schoharie County.

Traffic that travels across the backbones through Bob's house, but that is not destined for the KNOX node, travels from radio to TNC through the matrix, to TNC and back out the radio, without ever crossing that wireline link. The only traffic that goes through the wireline link is traffic that goes to the BOB node, the KNOX node, the NY10DX node or the CROWD node. This isolation between the node stacks allows the TNCs on each matrix to have less crowding in their communications between each other. More importantly the wireline link concept allows very large node complexes to be built without regard to RS-232 electrical considerations. A TNC is designed to hook up to one computer. Hooking it up to five other computers (TNCs) through the matrix is wonderful for building networks but stretches the capabilities of the TNC's RS-232 port. Because Bob wanted to have four backbone links, a Dx-Cluster port, a CROWD, a user port on two meters and an HF gateway, *plus* the BOB node he had to break up the node cluster into two separate nodes.

There is another advantage to having the wireline link between the TNCs: It makes a *real* nice light show!!

CROWD node

The first TNC on the left hand stack is the CROWD node. This TNC is plugged into the HexiPus™ matrix with four other TNCs but is not running TheNET software. Instead it's running NORD><LINK mini-conf software. It is fully TheNET compatible but instead of handling network traffic, the CROWD node's purpose is to allow stations who connect a round table conference capability. The name CROWD was coined by WA2WNI in 1988 and has since been widely used for this function. There are CROWD nodes at many sites in the north east. Each CROWD node has a different callsign and usually only one CROWD node will show at any node site. If you and a few friends want to have a conversation as a group you can all connect to CROWD at KNOX and each ham will see all of the text typed by each of the others. It's great fun and an excellent utility for emergency traffic handling.

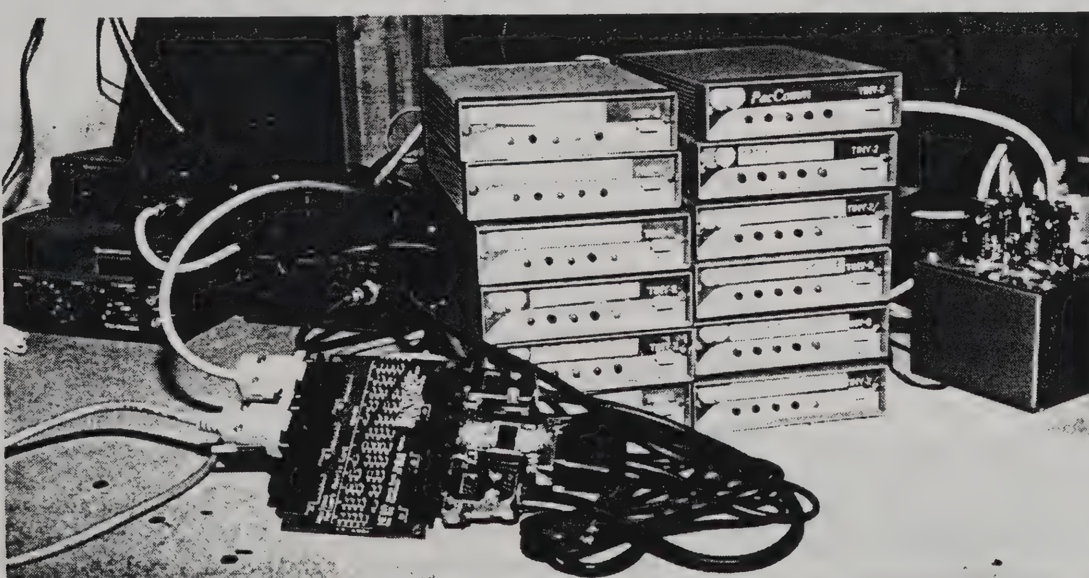
Simply connect to the KNOX node, then connect to CROWD. Now type slash W "/w" and the CROWD will give you a list of the other stations connected in and will announce your presence on the CROWD. If there is nobody on, hang in there and hopefully somebody else will check in during the two hours before you time out. If you type anything you reset the timer for another two hours. If you check in regularly you'll eventually start something and before you know it you'll have a nightly CROWD crowd. Also check out the CROWD nodes at STMFRD and CANDGA nodes.

TaddBench

The work bench that the node is set up on is a bench designed by KA2DEW. Currently there are eleven of the benches in existence. KC4ZWI (Pete) and KA2DEW descended on Bob's house last April and with Bob's help (plus about $230 wood and screws) built the bench in Bob's garage in about 13 hours. It took another 30 minutes to disassemble and re-assemble in the den/radio room. Now Bob has about 85 square feet of bench to play packet on! *(Bob also runs a satellite television system business and a VHF/HF station on the bench)*

Connect over to KNOX and play around with the R and N commands. For novices at TheNET network hacking you'll find the diagram very helpful for learning how those commands work and how TheNET networks are put together. Have fun!

—Dana, WA2WNI; Tadd, KA2DEW; Bob, WB2QBQ; and Bob, NQ1C (took the picture)



Close-up view of the KNOX node TNC stacks

Gracilus PacketTen Network Node

Copied from Quarterly #2.3

Hi, I'm a nodeop of a Gracilus PacketTen based node here in the Chicago area. Yes, we are aware of NEDA. We've talked with various NEDA guys at the Dayton Packet Dinner, and know a little about how you do networks. We basically agree with the provisions you make to rid your network of what you call HTS stations, so that the network can perform reasonably. Our attempts to do that here have resulted mostly in Chicago style politics by people from outside the Chicago area, rather than an acceptance of the work that has to be done to make things work and a commitment to get on with it. Congratulations on your organizational success.

As you may know, K9NG is a local here. The local group in its early days recognized the need for development, and supported these efforts. The result was a 9600 baud network, using NET/ROM, a few months after NET/ROM was shipping. I developed modifications that matched the K9NG modem to Midland 13-509s, and we installed a 9600 baud NET/ROM at the K9VXW-1 site and downtown Chicago. The downtown Chicago site uses, and continues to use, one of K9NG's original prototype modems interfaced to the Hamtronics 220 Rx and Tx boards, that he used for the tests published in his paper along with the FM-5. It's been in continuous operation since 1987 at that site. Reports of G3RUH's early efforts to do 9600 baud, as reported in the British Packet Newsletter electronic edition, and circulated around the Midwest by N8XX, were actually transported thru the operational K9NG Modem based NET/ROM network in the Chicago area.

We came to the conclusions regarding dedicated link network structure, about the same time the *staggered link* idea was published in GATEWAY, by the Florida guys. Considering it further, and the practical limitations of Midwestern access to high performance RF sites, we did the system design for an idea which I've dubbed ***Cellnet***. It's relatively easy to do networking in mountainous areas, where there are good RF paths at people's summer cottages, with minimal tower and feedline. It's a little tougher in the Midwest where the 50 Kbuck tower only gets you 30 miles, reliably, and donated tower space is not always forth-coming for 3 or 4 antennas to be able to build networks like NEDA does. Cellnet solves these problems.

Cellnet

With a single dual band antenna, or an up/down mount 430 and 2 meter antenna, 3 links and a LAN station can be operated. The links will be full duplex, dedicated point to point, which is about 6 times better throughput than the HTS protected, simplex CSMA technique used in the NEDA network, and for about the same cost. The PacketTen was developed in response to the Cellnet concept, but its application has grown beyond that. See my paper in the 7th ARRL Computer Networking Conference notes.

Recent History

Anyway, based on that prehistory, here's what happened about the time we finished up the main 220/9600 baud stuff in the area: N4PCR moved to town. Additionally, KA9Q's code was first being tried and the fact that it could handle so many ports, and simultaneous links made me think that it was time to start getting the Cellnet ideas I had down on paper. At Dayton that year, to my surprise, Karn and Dr. Death did a dog and pony show about a system very similar to the Cellnet ideas I'd had. So that really got me motivated to write it up and finish the system calculations. The excitement was contagious and N4PCR started working on a controller that would work with NET. His first attempt was good but just as he got it to work the 68302 chip became available. The 68302 is what the PacketTen controller is based on. He dropped the old project and began a 68302 project, formed a company, got two other guys to help him with it, and just started *really* working hard on it all.

Configuration

The PacketTen system has 4 building blocks:

- 2 versions of the processor card;
- and 2 versions of the interface card.

Processor Card

There is a PC plug-in processor and another version that can stand alone or be plugged into the PC plug-in version. Each processor card has the 68302 on it which has three DMA ports, a 68000 CPU and a RISC co-processor for fast port operation, all in one chip. Additionally each PacketTen processor card has an SCC chip on it for two slower speed ports (up to 19.2Kbaud). The 68302 ports can do up to a megabaud. The maximum throughput per processor is around 2 Megabaud.

Interface Card

The original interface board is a straight RS-232 port board. The next interface board they did is one that can handle the Kantronics versions of the standard 1200 baud and G3RUH modems right on the board. I think they might be coming up with a RS-232 interface board built as a plug in, with the same mechanical layout as the Kantronics modem, so they can do away with the original interface board altogether.

PC based 10 port switch

As I said, above, the two versions of the processor board can be connected. This makes a 10 port switch. The two processors, in such a switch, and the resident PC communicate via triple ported memory, for full performance between the three.

Software

The PacketTen comes with NOS software, ported to the 68000 in EPROM. A stand-alone PacketTen is the only NOS-in-a-box system available today. There is an

EEPROM for configuration memory. The KA9Q NOS has been expanded upon so that a NET/ROM-like user interface is there with full locked routes capabilities. The commands are different, being built into NOS, but the capability is there. I should know, I've been on his case to put them there, HI. The latest version of the code now has sorted NET/ROM nodes list too.

Defined Neighbors

By the way, we call locked routes *defined neighbors* in this part of the world so the NOS command that is used for that is the NET/ROM neighbor command. It is actually much easier to understand and communicate to others once you are up to speed with the Gracilus/NOS terminology.

Memory

The PacketTen has large memory for network node routing. It's much larger than the 32K of RAM that NET/ROM can use on a TNC2.

TCP/IP

Since the PacketTen is running NOS, TCP/IP goes right through it. There's no need for NET/ROM-to-IP gateway stations if two IP LANs are connected together through links made up of PacketTen stations. IP is truly the future for high performance packet. It will eventually have the capability of automatic routing through hierarchal and determined routes. This combined system is powerful enough that world wide real time routing would be possible since beyond the 'determined' routing horizon packets would be switched by hierarchies kind of like switching real time traffic like BBSs.

Now, for *non* real-time traffic. (zzzz)

—Don, WB9MJN@W9IUP

Full info can be gotten on the PacketTen from:
Gracilus
623 Palace St.
Aurora, IL 60506

Use of GE Phoenix VHF Radios for Node User Port Transceivers

Copied from Quarterly #2.3

Recently I have received a couple of GE Phoenix VHF transceivers that I considered for use at the CLV site. What a deal! A generous ham friend asked if they might be useful in our cluster. I saw an opportunity to reclaim a couple of expensive dual-banders and return them to normal use. These Kenwood stalwarts have given meritorious service for two years without a hitch. However, now was the time to replace them with rugged and durable radios.

The Phoenix is a 25 watt crystal controlled mobile unit whose small package lends itself well to stacking in tight quarters. It's tight front end and handy interface points make it a breeze to interface with TAPR clone TNCs.

A service manual was obtained and crystal data was determined. I ordered them from International Crystal Manufacturing Co. for about $38.00 a pair. Once inserted they tuned up on frequency without any trouble. The Phoenix seems to have a bottom frequency limit somewhere around 145.00 MHz. All coils were near the limits of their travel after tuning, however adjustments were positive with definite nulls and peaks where required. After about 1 hour tuning, the radio sprang to live with 23 watts output and a receiver sensitivity of about .35μV. Absolutely perfect for a rig used as user port with a low gain omni antenna.

I learned long that speaker audio is not necessarily the best for use with TNCs of the TAPR variety (or any other for that matter). I usually tap audio at the hot side of the receiver's volume control. This provides an easily found tap point and results in great audio with minimum distortion and filtering. Lo and behold, after studying the service manual I found an excellent audio tap right there on the Phoenix's rear interface connector. It is labeled 'FLTRD VOL/SQ HI'. As it turns out, this is audio after processing by a low pass filter that removes any CTCSS tones on the received signal. No problem for the TNC, as packet tones are not within this range. The filtered audio makes the job of the PLL much easier as it doesn't have to discriminated low frequencies and their resulting harmonics. An added bonus is that the volume control no longer has an effect on the desired packet tones and can be turned down so as to keep the site quiet and turned up when making antenna adjustments or to check the path.

The transmitter interfacing was straight forward. I simply connected the TNC transmit, audio and PTT lines to the radio's rear connector. Two audio inputs are available. I used the microphone input. I adjusted the TNC level and the radio's deviation control for 3KHz maximum deviation. This resulted in a sweet, easily decoded signal with no distortion.

The GE Phoenix appears to be an excellent radio for our purpose and could be for you too. They are easily found on the surplus market and require very little effort to retune for packet service. They are well built, durable and should provide many hours of dependable service.

—Charlie, N2CJ@WB1COY

SHERMN Node and Wireline Linking

Copied from Quarterly #2.3

The information presented on the facing page was transcribed from a block diagram and info/nodes listings supplied by Franklin. The existing setup as of the beginning of September at the SHERMN node did not include the wireline link between the two matrices. Rather all of the radio TNCs were hooked up in one stack. Franklin supplied the following Routes listing:

```
#GLIDA:N2JYG-11} Routes:
> 1 SHERMN:N2JYG-3 230 1
  1 WPADXC:N2JYG-6 230 1
  1 #GLIDA:N2JYG-14 230 20
> 1 #GLIDA:N2JYG-15 230 25
> 0 ERIEPA:NM3G-2 230 2 !
  1 #GLIDA:N2JYG-5 230 1
  1 #GLIDA:N2JYG-12 230 17
```

Franklin also supplied the information that he used to make the wireline link go. He said:

The TNCs for the wireline link are working on the bench and operating to full speed. The RS-232 port is 9600 baud as well as the radio port. TXDelay is set for 1 on the radio port. The EPROM was burned in as a full duplex TNC, Parm 33 set to 1. All other parms are standard backbone parms. The following mods were done to the TNC:

- Set radio port to 9600.
- Set RS-232 port to 9600.
- Cut the jumper on the modem disconnected header from Pin 17 to pin 18.
- Use a 3 wire jumper about 12 inches long.
- Connect Pin 17 on TNC A to Pin 18 on TNC B.
- Connect Pin 18 on TNC A to Pin 17 on TNC B.
- Connect grounds on the two TNCs together.

Franklin asked that I supply the info on connecting DCDs on the two TNCs as his setup didn't do this. Under heavy load collisions would occur without DCD hooked up. This is important. Also the LED for DCD won't work. Here is the required circuit:

Both Tiny 2 and MFJ:

1 Move the DCD select jumper to external DCD. Make sure the DCD light works once you are done!

2 Use a 74HC04 Hex inverter IC (74LS04 is OK substitute). Radio Shack part #276-1802 looks right. Make sure it's a LS or HC part though. You never know from those guys.

3 Connect pin 14 of the IC to +5 in the TNC. Pin 14 of another 14 pin 74 series chip in the TNC is a good place.

2 Connect pin 7 of the chip to Ground. Again pin 7 of another 14 pin 74 series chip is good.

3 Connect pin 5 of the modem header of TNC A to pin 1 of the chip.

4 Connect pin 5 of the header of TNC B to pin 3 of the chip.

Tiny 2 steps:

5 Connect pin 5 of the 5-pin DIN from TNC A to pin 4 of the chip.

6 Connect pin 5 of the 5-pin DIN from TNC B to pin 2 of the chip.

7 Connect pins 5, 9, 11 and 13 of the chip to pin 7 of the chip (Ground). This ties all unused pins.

MFJ 1270B steps:

5 Connect pin 5 of the 5-pin DIN from TNC A to pin 8 of the chip.

6 Connect pin 5 of the 5-pin DIN from TNC B to pin 6 of the chip.

7 Connect pin 2 of the chip to pin 5 of the chip.

8 Connect pin 4 of the chip to pin 9 of the chip.

9 Connect pins 11 and 13 of the chip to pin 7 of the chip (Ground). This ties all unused pins.

Construction method.

I must warn you that I'm a software weeny, not a technician. That's why I do newsletter editing and not PC board design and radio modification!!

What I do for a simple quickie operation like this is run the wires from the B TNC into the back of the A TNC through the TTL interface hole. Fan the 14 legs of the chip out flat and glue the chip to the top of the Z80 CPU or the 32K RAM chip. Tie the wires coming into the TNC to one of the legs of the regulator IC as a strain relief. Now run the wires to the chip.

Modification for using MFJs instead of PacComm. The only difference between the two brands that we are concerned with is that the DCD level is inverted. So the MFJ required an extra inverter stage on the 74HC04.

Now turn it on and let the smoke out!

Back to Franklin.

The only difference between the MFJ and the Tiny 2 is the port speeds are set for 19200 in the Tiny 2 and 9600 in the MFJ. Because I do not use Tiny 2s I am not sure of any other mods for that TNC but remember to do the standard mods for the MFJ, I.E.: Clock, U3 etc.

73's and type at you later, and pray for peace in the world.

—Franklin Werren - N2JYG
editing and graphics by KA2DEW

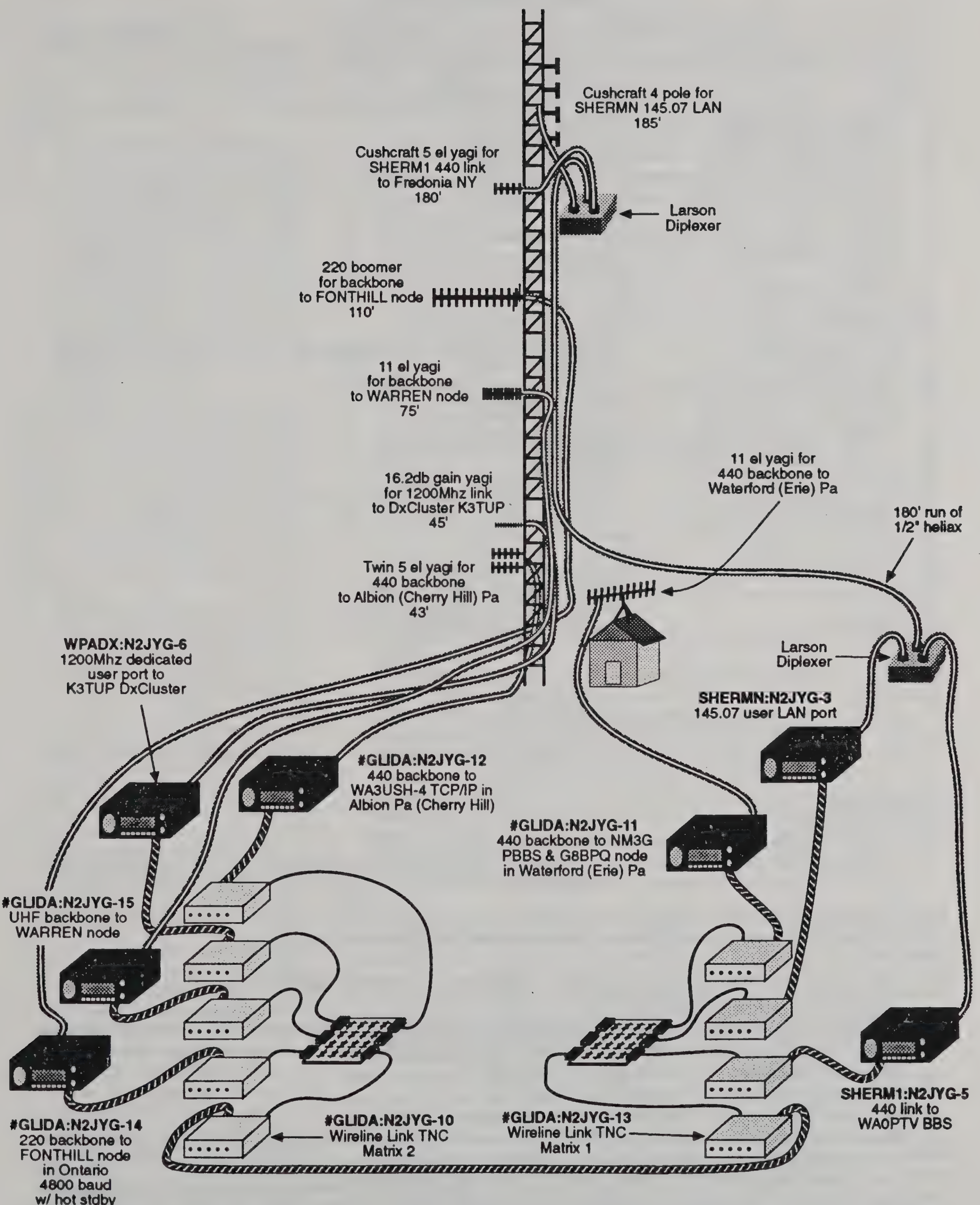


Illustration of the Sherman NY node owned by Frank Warren, N2JYG

NOS PARMS Table

Copied from Quarterly #2.3. This file roughly indicates the NOS equivalents of various TheNET PARMS. Please pay careful attention to the footnotes, some of them are important.

TN2.08		Description NOS
1	min quality for update	"netrom interface ax0 IPROCH 230" note 1]
2	HDLC channel quality	N/A (fits in with #1)
3	RS-232 channel quality	N/A (fits in with #1)
4	obso count initial value	fixed at "5" [note 2]
5	min obsolescence for broadcast	node is broadcast until it drops off
6	nodes broadcast interval	"netrom nodetimer"
7	FRACK	not sure
8	MAXframe (layer 2)	"ax25 MAXframe"
9	link-layer retries	"ax25 retries"
10	digipeat	"ax25 digipeat" [note 3]
11	validate callsigns	N/A
12	host mode connects	N/A
13	TxDELAY	"param ax0 1 15" [note 4]
14	broadcasts on/off	"netrom bcnodes ax0" turns on broadcasts for the interface named ax0
15	***pound*** node propagate	what does this mean?
16	connect command enabled	set through password file /ftpusers"
17	destination list max size	dynamically allocated
18	Time-to-Live initializer	"netrom ttl"
19	transport layer timeout	"netrom irtt" [note 5 - neato!]
20	transport layer retries	"netrom retry"
21	transport ack delay	"netrom acktime"
22	transport busy delay	"netrom choketime" (maybe? not sure)
23	netrom window size	"netrom window"
24	congestion control	(not sure)
25	non-activity timeout	n/a
26	P-persistence	"param ax0 2 128" [note 6]
27	slot time	"param ax0 3 xxx" xxx = slot ime
28	t2	(got to look this one up, book t work)
29	t3 timer	"ax25 t3 xxx" (zero means never")
30	N/A	
31	N/A	
32	N/A	
33	duplex	"param ax0 5 xxx"

One thing that NOS has that TheNet doesn't have:

netrom verbose [ON | OFF] if "off", on a broadcast the switch will only broadcast itself; it will not repeat its nodes list, only advertise its presence.

Disclaimers

(1) I didn't do it.
(2) This list isn't totally complete, so if you have questions **ASK**
(3) **Watch out for units**. For example, PARM 21 (ACKTIME) is in seconds, but in NOS "netrom acktime" is in milliseconds. This can be a serious problem if you've got retry timer units out of whack. Look it up.
(4) If you've got sysop privileges, you can remotely sysop these parameters, with the exception of those that can only occur once on initialization (like setting the incoming route quality)

Notes:

(1) This can only be done once, in autoexec.net. In the example, it sets interface ax0 as the node mnemonic "IPROCH" with route quality 230.
(2) In NOS, the obsolesence counters are decremented on a totally different timer than the nodes broadcast timer. Since this number is fixed at 5, if you want it to drop a node after 2 hours you would set "netrom obsotimer" to 2 hours divided by five, or 1440 secs)
(3) You may never digipeat "through" level three
(4) ax0 is the name of the NET/ROM interface. The example is 150ms.
(5) This is a really neat feature of NOS. The initial round-trip-time (netrom irtt) is the "default" time in which you expect an ack back. After a few packets, NOS starts remembering what the *real* time is. Thus, after it has been running for a few hours, it will learn to expect a response from a close node in a much shorter time than a far away node. A fudge factor is added to this round-trip time internally. Because of this feature, setting the transport retry limit to 1 is not practical; I have been using 3 and think that is fair.
(6) ax0 is the name of the interface we're programming. The '2' is the KISS parameter code for persistence.

—Chris Pigot, WZ2B

Installation of the MFJ 2400 Baud Modem in the Tiny-2/Micropower-2

Copied from Quarterly #2.3

While performing this installation, normal anti-static procedures should be followed. Avoid carpets and plastic chairs (plastic chairs are really bad) as a start.

1 Remove the nylon standoffs from the modem so it will fit on the TNC.
2 The MFJ2400 has provisions for an on board negative voltage regulator. A 79C05 (-5V, 3 terminal voltage regulator in a TO-92 package) should be installed at VR1 and the trace between pins 2 and 3 of CN2 should be cut and a jumper should be installed between pins 1 and 2 of CN2. This now allows the MFJ2400 to be powered from -12V rather than -5V.

Steps 3 thru 14 will have to be performed to get the TNC and modem to fit into the case, however I recommend getting it to work first by skipping to step 15 and simply plugging the modem directly onto the TNC's modem disconnect header (MFJ2400-CN1 to TNC-J5). Pin one at CN1 on the modem should be aligned with pin one at J5 on the TNC.

3 Remove the Connector at CN1 on the MFJ2400. This is necessary since the shape of the PCB does not allow it to be plugged directly into the TNC and still fit in its case. Be careful not to break any traces. Clear out the holes so wires can be soldered in place later.

To remove this connector, it may be easiest to break its plastic housing into small pieces with a wire cutter. The plastic housing usually can be pulled away from the PCB leaving only its contacts soldered in the PCB. These contacts can then be pulled out one at a time while applying heat where they are soldered.

4 Remove the connector at J5 on the TNC for the same reason as item #1 above. Clear out the holes so wires can be soldered in place later. To remove this connector, it may be easiest to apply heat to the solder side of the PCB at one of the pins. After the solder softens, pull lightly on the pin from the other side, it should pull out from the PCB and plastic. Repeat for all the pins.

For steps 5 thru 14, #24 gauge solid conductor wire is recommended. It is best to solder the wires in place on the modem first, with .75 inch of insulation from the bottom of the modem. Long stripped portions (2 or 3 inches) of the wire past the insulation will make insertion into the TNC easier.

5 Wire pad at MFJ2400 CN1 pin 6 to pad at TNC J5 pin 6.
6 Wire pad at MFJ2400 CN1 pin 11 to pad at TNC J5 pin 11.
7 Wire pad at MFJ2400 CN1 pin 12 to pad at TNC J5 pin 12.
8 Wire pad at MFJ2400 CN1 pin 13 to pad at TNC J5 pin 13.
9 Wire pad at MFJ2400 CN1 pin 14 to pad at TNC J5 pin 14.
10 Wire pad at MFJ2400 CN1 pin 15 to pad at TNC J5 pin 15.
11 Wire pad at MFJ2400 CN1 pin 16 to pad at TNC J5 pin 16.
12 Wire pad at MFJ2400 CN1 pin 17 to pad at TNC J5 pin 17.
13 Wire pad at MFJ2400 CN1 pin 18 to pad at TNC J5 pin 18.
14 Wire pad at MFJ2400 CN1 pin 20 to pad at TNC J5 pin 20.
15 Cut the trace between pins 11 and 12 of J5 on the TNC.
16 Cut the trace between pins 13 and 14 of J5 on the TNC.
17 Cut the trace between pins 17 and 18 of J5 on the TNC.

*The following traces should **not** have been cut on J5 of the TNC:*
Traces between pins: 1 and 2; 3 and 4; 5 and 6; 9 and 10; 19 and 20

Using the supplied cable which plugs into CN7 on the MFJ2400 perform the following:

18 Cut one of the connectors off the supplied cable, strip and tin the ends of the unterminated leads.
19 Solder the Orange wire to the negative (-) side of C24 (CN7-3 Ground)
20 Solder the Yellow wire to the 5V output (pin 3) of U5 (CN7-4 +5)
21 Solder the Green wire to the side of R6 which is connected to U15. (CN7-5 -voltage)
22 Solder the Red wire to pin 4 on the radio connector.(CN7-2 RX Audio)
23 Remove C27 from the TNC and solder the Brown wire to the C27 pad which is connected to R12. (CN7-1 TX Audio)
24 Connect a wire from R37 where it connects to T5 on the MFJ2400 (the end closest to IC6) to Ground (Modem enable wire). Grounding this wire enables the 2400 baud modem.
25 The Radio baud rate setting on the Tiny/Micropower should be set to the 2400 baud position
26 Place some insulating material between the Modem and the TNC to prevent shorts. (note - anti-static foam is not an insulator. I have seen people use it as such which causes some real interesting problems)
27 Clip approx 1/8 inch off the ends on the pins at CN6 on the modem. They are a little long and will short to the case unless cut.
28 T3 should be bent over so its tab does not short to the case.
29 To hold the modem in place, a couple of pieces of buss wire can be wired from the ground plane on the component side of the modem along the edges (scrape away the solder mask) to the ground traces along the edges of the Tiny/ Micropower.

This completes the installation.

NOTES:

To restore 1200 baud operation, first the Modem enable wire connected to R37 on the MFJ2400 should be disconnected from ground and be left to float (Item 24). Second C27 needs to be reinstalled (Item 23), but the Brown wire need not be disconnected since the 2400 modem goes to a high impedance output. Finally switch the radio baud rate setting back to 2400 baud. Proper wiring of a 3PDT switch to perform the above would allow switch selecting 1200 or 2400 baud operation.

For proper 2400 baud operation thru the radios microphone jack proper setting of the transmit deviation is necessary. 3.5 kHz is recommended. The transmit audio level can be adjusted by using the R12 on the TNC and if the output range is not sufficient or touchy the jumper can be moved on CN6 of the MFJ2400 or change the adjustment range (see MFJ2400 baud manual) The potentiometer on the MFJ2400 can also be used to adjust the transmit audio level as well.

The TNC which originally would function with input voltages from 9 to 14 volts will now only work with 11 to 14 volts input. This is because the negative voltage picked off of U15 on the TNC will not be enough for the -5V regulator on the modem. When this happens the modem stops functioning although the TNC will still function normally over the RS-232 port.

—Bill Slack, NX2P

NEDA and Servers On 2 Meters

Copied from NEDA Quarterly #2.1

This article addresses the question of "What is NEDA's stand on servers using 2 meter user ports to access the NEDA network". A server is any station that is on the air as a service to users other than it's owner. This includes PBBSs, DxClusters, TCP/IP gateways, DOSgates, CD-ROM callbook servers and any station that sources large volumes of data to other stations across the network.

NEDA network participants voluntarily agree to a consistent set of technical guidelines. These guidelines only specify the software running in the node TNCs at each NEDA node site and the interlinking methodology between nodes. However, at least one of the club's technical documents describes how detrimental a station sourcing high volume data to a user port can be on users. Let me restate:

On a given node if the server has it's own uplink on 440 and the users access the server via the 2meter user port there will be very few collisions even when loading is at maximum.

If the server is accessed via the 2meter port (the server is on the same 2meter freq.) then there will inevitably be collisions. If the user port is fully loaded by users accessing the server or by other server activity then there will be lots of collisions and efficiency will be no more than 19% and sometimes as bad as 0%. [1]

Server ops who do tie into their local network on 2 meters will definitely degrade the performance of users at that 2 meter port. The users will also hamper communications of the server with the network. If the server were to find a dedicated access into the network the server and users would both benefit in efficiency, and more fun would be had by all. While servers may share a frequency with users, functionality will be far below that of servers on dedicated links to the network. If the server is offering a valid service there is no reason in the world that those benefiting from same couldn't help fund the dedicated link for the server!

It is up to the node sponsors to determine their own policy to approve or disapprove of server activity on the frequency of the node's user access port. It is up to the potential server's operator to respect the wishes of the sponsor of the node. This policy is no different that the manner in which voice repeaters are operated.

The more fun that is had and the more efficiency with which the network is run, the more the network will grow and the more different kinds of services will be available. The bigger the network is, the more good people there are working together for a common goal. That is NEDA's position.

[1]*Binder R. Abramson, N, Kuo, F., Okinaka, A., Wax D. (1975), 'ALOHA packet broadcasting - a retrospect', AFIPS Conference Proceedings, 44, 1975 NCC, 203-215. ** reference and information source for the Quarterly was taken from "The Data Ring Main" by Flint. Published by Wiley.*

—The NEDA Board of Directors

Notes about the 9600 G3RUH modems
(including the NB96)

Copied from Quarterly #2.3

You may or may not have known that voltage variations on the 12V line affect the modulation level out of the NB96. This has caused some problems in some of the stuff I have done. A voltage change from 12 to 13.8V can easily change the transmit deviation (when fully interfaced to radio) from 3KHz to 5KHz. This is something to be aware of when setting up the 9600 BPS modems

I heard a report about a fix, which has made a big difference for some UoSAT ops, but with no cause. I looked into it, and I can see why. Here it goes:

On the NB96 card (internal or external) a reference voltage is derived off the 12V power supply by a resistor divider. This divider consists of two 100K with a .1uF capacitor from the junction of the resistors to ground. The resistors in question are RS1-3 and RS1-4. This provides a 6V reference. The capacitor and resistor values set a time constant of .005 seconds which filters out high frequency noise (above 200 Hz) which may be present on the 12V line. The 200 Hz significantly below the modulation rate so that is good, but it is too high to provide isolation from 60 or 120 cycle ripple from a poorly regulated power supply, and it provides no isolation from the low frequency voltage changes which occur when the transmitter keys up. I have noted that I have has to set TxDELAY much longer than I would have expected in some installations. This may be the root cause of that.

That resistor divider is buffered thru a op-amp follower to convert it to a low impedance suitable for use by the remainder of the circuitry. This is good or it would be much worse, but still any fluctuations in the reference on the high impedance side of the op-amp will be faithfully duplicated on the low impedance side. Since that reference is used by all the analog circuitry, and noise there degrades the overall system performance.

The fix to this problem is to replace RS1-3 with a 6.2V zener diode (anode goes to gnd). I would recommend replacing RS1-4 with a lower value resistor to increase the idling current of the zener minimizing voltage swings. Something between 1K and 5.1K should do nicely. Also adding a larger capacitor in parallel with C24

Continued ➡

Tiny 2 TNCs at 38.4Kbaud

Copied from Quarterly #2.3

I did some tests upping the clock speed on the Tiny-2 TNC. I successfully got them to work with a 9.825 MHz clock which is 2x the normal clock. Since the baud rate clocks are also derived off of the main system clock, the 19.2 position now generates 38.4Kbaud. As higher speed links are more common, and the number of these links at one site increasing, the normal capacity of the 9600 or 19.2K RS-232 link between the TNCs is becoming increasingly taxed. 38.4 may be a viable solution.

One of the driving factors behind the test is the poor high speed performance of the standard TNC code. Doubling the clock speed should go a long way to helping. Of course all the timing parameters are now only one half their original value.

A TNC modified as such does not have the same computing power of a "data engine" or PC, however using it for high speed packet does become viable. Can you call 38.4 high speed? Some people seem to thing 56K is a magic number. Sort of like 9600 is magic compared to 4800 although 4800 is quite respectable. Of course some people would say anything less than a 1 megabit is low speed. In my mind, 300-2400 is standard (slow) speed packet. 4800-19.2 medium speed, and I would call 38.4 to 1 Mbaud high speed. Above that can be very high. All is relative.

Here is an idea. The G3RUH modems can operate above 9600 by changing some of the filter components. In fact I am told they would work much higher including 38.4K. So now we have a TNC with a 2x clock, and set the radio baud rate to the 19.2 position so it is actually operating at 38.4K We modify the G3RUH modems filtering so it will do the 38.4K. So far so good, the proceeding is pretty straight forward. Next we take one of the Tekk KS-900 data radios. We change some of the front end filtering on the transmit side so it will pass higher frequencies than it was originally designed for. Now the only limiting factor is the 20KHz IF filter of the radio. We need more receive bandwidth. It is relatively easy on the transmit side, but we need to find a wider IF filter. Well I was thinking, what about FM broadcast receivers? They must use a wider IF and in fact, it should be about what we need. Perhaps a bit wide, but it should do it. What do you think? Basically a 38.4K link for the cost of a 9600 baud link plus a few extra parts and some work. Of course with the wider bandwidth range would be reduced. I am making some assumptions about the ability to modify the radio which is where I am not sure about. Any comments?

> *All the new Tiny-2s use 6MHz parts, but older units may use slower speed parts which means they could not be modified with the higher speed clock. In fact there is no guarantee that new units will work at the high speed. I only made the test on two units both of which worked fine. Also after upping the clock speed, tests should be made with the TNC at various temperatures to make sure it will function reliably.*

—Bill Slack, NX2P

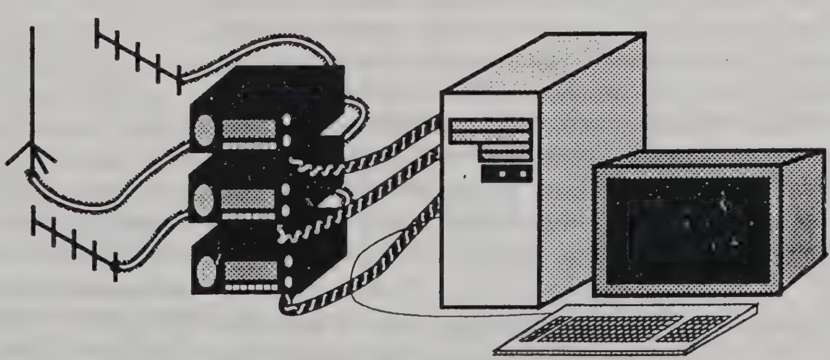
[editor: Bill runs a four port ROSE switch site from his house in Sparta NJ. Bill is active in promoting packet in his area as well as running a custom PC board design business and being a PacComm dealer and a dad!]

---

would be a good idea. A 5.0uF capacitor would maintain the same time constant if a 1K resistor were used instead of a 100K. 25uF would provide some immunity to 60 and 120 cycle stuff, but that starts getting big.. The Zener works well on the lower frequency stuff so the time constant between the resistors and capacitor is much less important so it is not worth getting too hung up on the capacitor. C24 should be left in the circuit as a .1uF. That provides isolation from the high frequency stuff which the Zener is not good on. RS1-3 and RS1-4 are parts of a resistor SIP network so it is not simple to just remove these without affecting the other resistors. Since the impedance of the reference circuit is being dropped so low with the above changes, these resistors can be left in place with little effect.

Note: the above note/modification not only applies to the PacComm units, but probably all the G3RUH based modems since the above problem is in his original design.

—Bill Slack, NX2P



The Exposed Receiver Syndrome

Copied from Quarterly #2.3

By now you have probably heard of the Hidden Transmitter Syndrome (HTS), and NEDA's policy of insisting that all backbone links be HTS free. The hidden transmitter problem is the scourge of low cost backbone attempts. A related but less well known problem, which can have a detrimental effect on hilltop User Ports, is the Exposed Receiver Syndrome (ERS). The Exposed Receiver is the receiver at a User port that hears much more than it needs to; it is "exposed" to signals working other nodes in distant communities.

The problem with nodes experiencing ERS is that User port TNC defers to the distant signals, waiting unnecessarily before sending. The TNC does not realize that these other signals that it hears are so distant that it could safely transmit to its local users without causing interference to the distant stations. In the meantime, the local users' TNCs may retransmit because of the unexpected long wait for an acknowledgment. This further adds to congestion on the channel and slows things down.

What does this mean for you, the operator (or future operator) of a NEDA node? First, this shows the need for small, local coverage nodes ("nodes in homes" as Tadd calls it). As activity grows, the packet network will benefit from a "cellular" approach, where several local coverage nodes are linked together via a HTS backbone, rather than trying to cover the same area with one massive coverage mountaintop node. Second, the User port frequency should be chosen to reduce ERS from other nodes on the frequency. This means that wide coverage area nodes should be on less popular frequencies, as opposed to say 145.01. High power output and high gain antennas may actually hamper the usefulness of your node rather than enhance it. If the node is not centered on the intended coverage area, then use a directional antenna to target your audience and limit your exposure.

Phil Karn, KA9Q, proposed a software solution to this problem in a paper at the last ARRL Computer Networking Conference. This may be a long term solution, but it would not be compatible with existing systems now.

Consider adding a CTCSS encoder/decoder to the node. If 2 nodes sharing the same frequency each had CTCSS encoders, they could each sense the presents of the sub-audible tone from the other node and then squelch the receiver. (The sense of the squelching is the reverse of what is usually done with CTCSS). When a local user started transmitting he would capture the receiver at the local node, thus ending the sub-audible tone and allowing the receiver to open. Since each nodes' receiver would stay squelched whenever the other node transmitted, the nodes would never end up waiting for each other before starting to send.

—Rich Place WB2JLR

Split UHF Frequencies for Maximum Band Utilization

Copied from Quarterly #2.4

At the VE2RM:WQC node site I have installed a packet repeater. The repeater site is operated by the VE2RM radio club. Because of the split frequency (5MHz) nature of the repeater I was limited to installation of UHF links. I proposed that instead of running point to point backbone links on UHF simplex frequencies, which are scarce at the site because of the repeater, that we run our point to point links on half-duplex channels whose pairs are adjacent to one another. Thus as the repeater transmits on 446.025 and receives on 441.025, the links would all transmit in the region of 446 and receive in the region of 441. This implies that the sites we're linking at must also be using a split, half duplex, method of UHF channalization. This plan allows for many UHF links in and out of the same site.

The idea has been implemented now in both the Montreal and Quebec metro regions and seems to be without flaw.

—Burt Lang, VE2BMQ

G8BPQ/Hexipus Problem Note

Copied from Quarterly #2.4

For all versions of G8BPQ you must run the port in half duplex to get it to talk to the active coupler and Hexipus. This is not documented correctly in BPQ until version 4.04.

-K1MEA, Jim Wzorek

Nice and Direct

Copied from Quarterly #2.4

This is the info text on the Watertown NY node.

```
WATERT:WB2QJO-5} }
Watertown NY in Jefferson Cty
144.97 user LAN!
NO DIGIS OR SERVERS ON 144.97 PLEASE!
Set your DIGIPEAT OFF
Talk to KB2DAJ about adding to the network
```

Screamers!
or
The Network ... What is it?

Copied from Quarterly #2.1

How many times have you tried to explain to a newcomer what the Network is all about and how NEDA's version is superior to a string of digi's or a single frequency arrangement of nodes only to have that person walk away from you after 5 minutes or go glassy eyed and mumbling to themselves. We all have at one time or another and it hurts to realize that many packeteers have little or no idea of what goes on in the network. If you think about it for a moment it isn't that much different than people's approach to the phone system in that everyone uses it and can detect the slightest decrease in performance but not many understand how it works. Next time you try to explain the system's workings to someone think about how NEDA evolved and your explanation will make far more sense.

Going back to the earliest recorded history of packet we find reference to a Neanderthal race of pseudo techs that developed a method of communications that consisted of placing individuals, one after another, on a string of mountaintops. Each was within earshot of the next and each individual was equipped with a magaphonelike gourd that was used to enhance his directivity and to increase his unidirectional gain. Each person was capable of repeating what he heard but only the first and last had a check back memory, intelligence was a rare commodity in those days. Messages would move from one end of the mountain chain to the other through a series of shouts and grunts one to the next ending hopefully with a final shout to the end mountain. What with screaming beasts in the jungle below, erupting volcanos and violent electrical storms the message was often lost somewhere between start and finish. This would result in a series of screamed "What did You say?" — "Beats me, let me check with the next guy "exchanges that would eventually return to the beginning Mountain top and then the process would repeat itself. If one was patient eventually all pieces of the message would arrive at the end mountain and communications was deemed to have taken place. The age of the DIGIPEATER had come and communications limped along for several years in this fashion.

For some this approach seemed to be a little clumsy and finally the question was asked "What if each person on each mountain top had a check back memory?" The mountain tops quivered as this flash of brilliance reverberated through the system. "With individual memory we wouldn't have to restart missed message segments from the first mountain but rather we could correct mistakes or repeat portions missed between any two mountains at the point where the problem occurred." It didn't take long for people to realize that this was a vast improvement over the aging DIGIPEATER approach. One local leader of an especially primitive tribe of people in what is now NY State was heard to say "I Node There was a better way to get thin dun" and thus the Age of the Node was born.

The new system grew in leaps and bounds and soon there were nodes all over the place, all screaming and grunting from there respective mountain tops at the top of their lungs and remembering things between screams and passing traffic hither and yon. As time when on more and more nodes appeared people started to notice that messages were taking longer and longer to get from one end of the system to the other and once again people started to wonder if there was a better way to get things done. A young chief with long flowing golden tresses journeyed to the top of one of the mountains and after listening for a few moments to the cacophony of grunts, screams, hiccups and burps exclaimed "This noise is awful, I can't tell one message from the next! We NEDA better way of doing things". Yup, you guessed it. This was the start of the NEDA Network and it represented another great leap forward for communications.

The young chief reasoned that the problem with the Old Node system was that with so many Nodes on so many mountain tops all screaming at the same time individual messages were being lost in the roar. He returned to his village and in a stroke of genius discovered that certain member of his tribe could only hear low tones while others could only hear middle tones and some could only hear high pitched tones. He then selected other members of his tribe with low, medium and high pitched screams and he started to build his new network. He placed pairs of screamers on mountain tops with different pitched voices and along side of them placed pairs of tribe members with different hearing responses. He found that by using this approach they could be receiving a message form a distant mountain and at the same time be sending one to the next mountaintop without the two processes interfering with one another. Once again the communication rate bounded upward.

Well many years have since passed and the young chief is now old but his work goes on. Though careful selection some mountain tops have 6 or more pairs of screamers and listeners. It is rumored that some mountain tops are populated with FASTER screamers and messages are moving more rapidly than ever....

I hope that this historical perspective helps you the next time someone says "Tell me about NEDA".

73's

The Old Packeteer

Problems and some fixes for TheNET

Copied from Quarterly #2.4

I think that there are three things wrong with the node networking software systems.

Problem No. 1 is that when you send data into one of these nodes, the node will not return an acknowledgment to the originating station of the packet until the information has been sent to the next node. Let's take a case where the node is relaying the information to another node or user. To visualize this, say that you have your local uplink LAN port into the network. When the data is received by your local LAN, that node first sends to the next node in the path to get to its destination, waits for an acknowledgment from that next node and *then* sends you back your acknowledgment.

Problem No. 2 is that when you connect to a distant node and ask for a nodes listing, the entire listing must be received by each node in the connected path before it will be relayed to the next connected node, or end user. To visualize this, lets say that you connect to your local LAN node, and then connect to another node in the network that is out of your local area and that there are a dozen other network nodes in the path. When you type "N" (for example) the information will start making its way across the network to the node at which you made your entry into the network, but you do not receive any information from your local node until the entire listing has been received at your local network node. This means that if the path was working and it failed for some reason, you will not even get a partial listing.

Problem No. 3 is when you connect to your local uplink node and you wish to connect to another node in another state (for example) and you issue a "C xxx" your packets are essentially digipeating from the node at which you uplinked the destination node. But, how can this be? We have been told that to avoid this is why the nodes were developed in the first place, right? Right. The problem comes from the way that NET/ROM, TheNET, and TheNET-Plus handles the level 3 and 4 connections. On a level 3 connect between nodes you *do* get node-to-node-acknowledgments, but on level 4 layer connections you're essentially digipeating across the network and backbone networks to your destination. This is one of the reasons why node hopping across the country will often be so difficult. If there is local activity that is keeping the network busy in part of your path and your packets are trying to traverse that part of the network, it will be competing with the most aggressive timing parameters on the local connections. Another factor that plays here is the timing parameters. Nodes operate the same way your TNC operates, with parameters like FRACK, PPersistance, DWait, and RETRY, and if your Packets are 'digipeating' across the network in a level 4 connection these parameters might well time out your link connections before the data has even had a chance to be relayed back to your last node connection. This also is true for those BBS stations that advertise "xxxBBS". If you should connect to your local node and see a "xxxBBS" in the nodes listing (by typing "N") and see a BBS that you'd like to connect to but it is another state or even another part of your network, there might be a dozen or more network nodes in the path to get to that station and you're essentially digipeating the entire route. It would probably be best if the "xxxBBS" nodes were to be limited in the network to the area of intended coverage anyway and not propagating throughout the network and across the states, but that is another story.

Best bet here is to 'stage' your connections. A knowledge of the network map is useful. You can also figure out the network your self by stepping through it if things are set up that way. For instance. First connect to the local LAN node, then type "N xxx" where 'xxx' is the destination node. The node will then reply with any information that is available to get to the destination node, such as; (see side bar!)

In this example I only ventured but a short distance, which never left my house, but this very same method has been used for years to travel across the country from one state to another several thousands of miles away! While I was living in San Jose, California I used to be able to connect with nodes in South Dakota! Now that I'm again living back in Washington state I still get connections from N7OO using various VHF and HF links from Sierra Vista, Arizona.! You'll find that there are places where you can skip several nodes in your connect path over a period of time, and that there are others that you must connect to get past a place that has poor propagation conditions or heavy loading from BBS or user activity, but that your over-all ability to get from one place in the network to the other will be vastly improved.

Disclaimer: NET/ROM or its equivalents are what we've got to work with, there may be other networking software packages out there that operate in a different way, but that doesn't mean that this software is inferior. However, knowing the limitations will better enable you to be able to use the software more effectively.

—Scott Kronk, N7FSP

```
C N7FSP-14                                  - entered to connect to uplink node
Connected to N7FSP-14                       - Connected!
N ALKI                                      - entered to the local uplink node
*ALKI:N7FSP-14} Routes to: ALKI:N7FSP-1     - this is a reply from the node
> 190 6 0 WSEA                              - second line of reply from node
C WSEA                                      - next node in path to destination
*ALKI:N7FSP-14} Connected to WSEA:N7FSP-5   - Connected!
N ALKI                                      - entered to the local uplink node
WSEA:N7FSP-5} Routes to: ALKI:N7FSP-1       - this is a reply from the node
  255 5 1 ALKI                              - highest quality to destination node
  254 5 1 *ALKI2                            - possible back-up route
  254 5 1 *ALKI3                            - possible back-up route
C ALKI                                      - connect to highest quality path
WSEA:N7FSP-5} Connected to ALKI:N7FSP-1     - Connected!
```

LAN Architecture or Should I use a Beam or an Omni?

Copied from Quarterly #2.4

There has been controversy since the early days of amateur packet radio as to whether a packeteer should use an omni or a beam. I'll try to resolve that in this article. I think that I can show that in modern metropolitan packet radio a user station should utilize a beam if possible.

Most packet radio operations in the U.S. occur on 2 meters. In most situations when a packet user turns on the radio and TNC the station will hear other sites. Some of those other sites will hear yet other sites and so on. In most cases there will be more than one server, node, digipeater etc. on the frequency. This is far from ideal. In this case planning either has not taken place or has not been effective. For the purposes of this article I'm going to focus on LAN channels where planning has taken place and where we're now trying to make it effective.

Fixed # of stations, all stations hear each other

There are two LAN architectures available to users of current day off the shelf TNCs. The first is the same architecture used in commercial CSMA ethernet systems. In this all stations can hear each other. All are basically omni. All have equal priority and may make a local decision on when to transmit and be pretty sure of not colliding with another station.

This kind of LAN is possible on Amateur Radio only where spectrum space is not a premium and all of the packeteers are in a planned region. This may be the case in a small community, not a major metropolitan area.

One server, stations don't all hear each other

The second architecture is one in which it is not possible to predict how many active stations can hear each other. Using standard TNCs the only form that this LAN can take and still function with better than 20% efficiency is the form in which

- one station on the LAN can hear and talk to all of the other stations
- that one station is the only server on the channel.

These two points are usually the case on *designed* LANs because all of the users access one node or one server on a given frequency.

It can be proven experimentally or via simulation that the only way to efficiently use a CSMA system with hidden transmitters requires that the total utilized channel time must be less than 20% of the available time. If the server is not a hidden transmitter to anybody then it may use as much time as it wants. Only the user stations need divide the remaining time by 5.

In this scenario a beam should be used for all user stations if possible because

- it won't affect the channel utilization calculations at all whether the user stations can hear each other or not, so long as the user stations don't transmit very much
- and the geographic coverage of the LAN may only be controlled if the individual user stations cooperate by using beams.

The two drawings show geographic area used by a LAN where the users have beam antennas and by a LAN where the users have omnidirectional antennas.

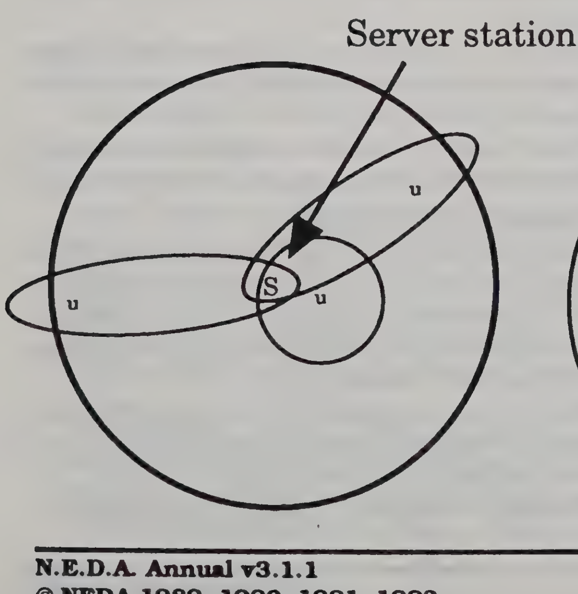
It can be seen from the drawings that if a *large* group of people operated to a server station, using omnis, that the size of the LAN would be around twice as wide as the LAN where user stations are operating with beams. This would lead to one quarter then number of possible LANs on each 2m frequency in the metro area.

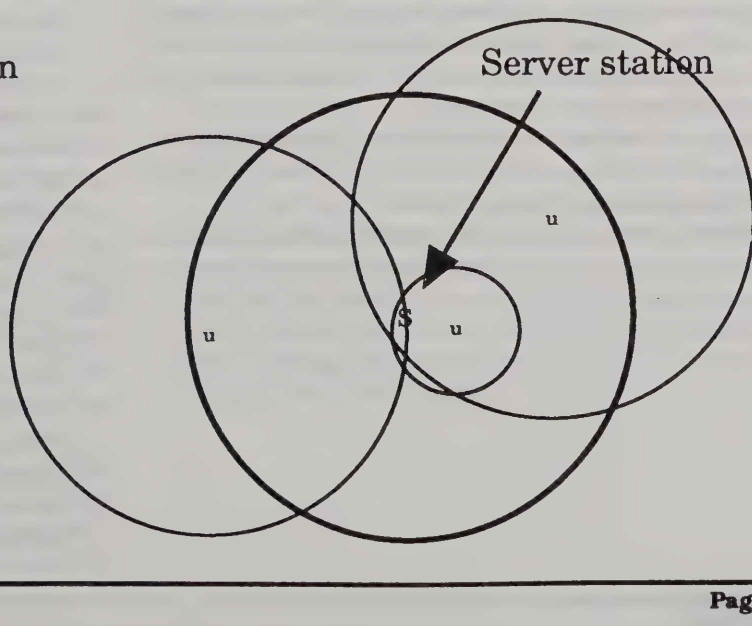
Since the most efficient (highest data bandwidth) utilization of a frequency, with a given baud rate, is with a single-server type of LAN, the amount of data bandwidth available on a frequency would go up by a factor of 4 if all user stations used beams. (Area of a circle is πr^2)

Since most 2 meter LANs are not planned there is an immeasurable improvement to be gained by running cellular LANs, with users having beams.

Your next phase, after you prove out the cellular concept is to start reducing the size of your LANs. By reducing the max number of stations on a LAN you increase the performance you give each station. Baud rate is secondary. Obviously having a digipeater on the highest building in the city is right out (hi).

—Tadd, KA2DEW





NEDA Constitution

1. Purpose of this Article

a. This article lays down the rules for operation of the North East Digital Association. No other N.E.D.A. document may change or replace the rules set down in the Constitution. The Constitution may only be modified by the procedures described herein.

2. Officers

a. There are six Board of Directors positions plus appointments and alternates. The board of directors are elected for two year terms. Three of the directors are elected annually.

3. Appointments

a. Appointed positions include Treasurer, Chairman of the General Meeting, Membership Director, Board Member Alternates, Chairman of the Technical Committee and Network Regional Sysops. The Network Regional Sysops report to the Chairman of the Technical Committee and are considered members of the Technical Committee.

b. Other appointments may be made at the direction of the board of directors. These appointments are made by the board of directors. Only voting members may be appointed to a committee chairmanship, board member alternate or office position. Board members may also serve other appointed positions and appointees may serve multiple appointments.

4. Board Member Alternates

a. Each board member may appoint an Alternate to represent him or her at board meetings in the event that the board member is unable to attend. The Alternate must be approved in advance by the board during a board of directors meeting in which the board member presenting the candidate for Alternate must be present. Appointment of an alternate may be discontinued at any board meeting at the request of the board member the alternate represents, or with a majority or tie vote of all the board members.

b. The Alternate has full voting rights at board meetings in the absence of the board member which he or she represents.

c. If the service of a board member that an alternate represents ends the alternate position is also ended.

5. Removal of a Person From Office or Revocation of Membership Privileges

a. A petition for removal of a person from office or membership must be submitted in writing to the board of directors with a minimum of four signatures of voting members. The petition must be presented at least two weeks before a quarterly board meeting in which it is to be acted upon. The board of directors must vote on the petition at a quarterly board meeting. The document will be kept in the club archives unless removed and expunged at a later board meeting.

b. This person being removed is held as a removal-pending member for one quarter and then is reviewed at the following quarterly board meeting. This issue is then presented in the minutes in the Quarterly so that it may be reviewed by all the membership and commented on before the following quarterly board meeting.

c. A person removed from membership is not eligible for voting membership unless the privilege is restored by an act of the board of directors at a later date.

6. Membership

a. Membership is open to all. Dues are at least 2 levels for individuals. One of these levels is called Voting Membership. Voting membership is open to all except as defined under 'Removal' above.

7. Dues

a. Dues are paid to the Membership Director or his designee who then forwards the funds to the Treasurer. Dollar values of dues is set in the NEDA bylaws but the dues level for a Voting member is $25 or greater. Dues are used to fund:

- operating expenses for the club;
- development costs for club products that facilitate network growth.
- documentation in the form of an Annual and Quarterly
- documentation in the form of free technical documentation distributed for the benefit of packet networking.
- documentation in the form of free promotional literature on NEDA and on packet networking.

8. Membership Privileges

a. Voting Members receive the 4 copies of the NEDA Quarterly per year and a copy of the Annual each year. The Annual is delivered to the member at renewal time (after renewal) or at the anniversary of the member's membership.

b. Voting members are invited to attend the Board of Directors meetings, run for office annually and vote for officers by mailed ballot.

c. Additional privileges are defined in the bylaws.

9. Board Meetings

a. A Board of Directors Meeting is a physical gathering of the board members.

b. A minimum of 4 directors must be present to open a board meeting. The board meetings must be announced via the NEDA Quarterly or via packet mail to every voting member at least two weeks before the meeting. If a quorum of board members is not available to start the meeting a new meeting must be scheduled and new announcements must be sent.

c. Board meetings must be held in different cities each time to make it possible for all voting members to have equal access to the proceedings of the board of directors.

d. Board meetings may be attended by voting members or those given special dispensation by the board of directors or any approved by the bylaws.

e. Board meetings must be held 4 times per year. The 4 quarterly board meetings are held as close as possible to the first week of January, April, July and October. Additional board meetings may be called by the board of directors with a vote of 4 board members. A board meeting is required in order to:

- spend club funds.
- discipline a member;
- change the appointment for network sysop or chairman of any committee.
- re-assignment or assignment of a board member alternate;
- change the constitution or bylaws
- appoint the chairman of the annual meeting or change that appointment.
- form or disband any committee.

f. Actions which must occur at the board meeting include the reading of a current NEDA treasury report. This will be recorded in the minutes and printed in the subsequent NEDA Quarterly.

10 <removed>

11. Elections

a. Elections are held by mailed ballot after the October Board of Directors Meeting. Immediately after the October Board of Directors Meeting attendance of each member, over the previous year's board meetings, are tallied. Any voting member who is paid up for two years from the end of October of the current year, who has attended half of the year's board meetings, and who are not already in the middle of a two year term are automatically nominated and are listed on the ballot.

b. This ballot is sent to all NEDA voting members complete with a self addressed stampled envelope. The envelope also has a return address label with a note stating that the return address must be filled in for the ballot to be counted. The ballot includes instructions that the voting member should order all of the listed people in ascending order, 1 for first choice, 2 for second choice. This way the results will still be meaningful if one or more nominated members are unavailable to fill the positions. The ballots are mailed to the club POBox and then counted by the recording secretary or one of the board members whose term is not expiring this year.

c. The ballots must be mailed out to all NEDA voting members within two weeks of the board meeting. They must be returned to the club POBox within five weeks of the board meeting. Results are included in the Quarterly or are mailed out separately to all members to arrive at least a week before the winter board meeting.

d. The results include the following statistics:

- total number of ballots sent;
- total number of ballots returned.;
- list of all nominees;
- list of the three new board members;
- and a list of nominees who abstained but who had a higher vote than the selected board members.

e. If three new board members are not chosen by this process then a board member may be chosen by consensus of the founders and the existing board from those voting members who were previously board members and who ended their term as board member in good standing. If there still are not three eligible new board members then the club must be dissolved.

12. Board Member Responsibilities

a. Board members or their alternates must attend the quarterly board meetings or obtain an alternate to handle meetings the board member cannot attend. Failing to do so twice in a single year is grounds for removal from office. Board members or their alternates are also obligated to attend additional board meetings called by verbal agreement by any four of the board members.

b. Board members represent NEDA and are obligated to carry out the NEDA Charter in regards to dealings with other members and non-members.

c. The board of directors as a body are responsible for seeing that the NEDA Quarterly and the NEDA Annual are published on time. As these are the instruments of the club and as the NEDA Quarterly is the means by which the financial operations of the club are published to the membership, the paying membership has the right to expect these documents.

13. Filling Spots on the Board Due to Board Member Resignation

a. If a board member resigns or is otherwise no longer available to fulfill the remainder of his or her term a new board member is selected to serve until the next annual meeting. The new board member is selected from those voting members who were previously board members and who ended their term as board member in good standing.

14. Network Maps

a. Network maps must be maintained and are presented in the Quarterly. The maps must consist of at least the callsign, nodename, location (at least relative), user access frequencies for AX.25 (if any) and backbone connectivity for all NEDA network nodes.

15. NEDA Quarterly

a. The NEDA Quarterly is published within 60 days after the quarterly board meeting. The Quarterly is fully described in the bylaws but as a minimum must include the minutes of the board meeting (including the treasurer's report), the network maps, and membership roster.

b. The board may delegate the task of production and mailing of the Quarterly but maintain the responsibility.

c. In the Fall edition of the Quarterly whatever results that are available from the annual elections are printed. This may include the nominees or the final results.

16. NEDA Annual

a. The NEDA Annual is the current statement of NEDA packet network involvement. This includes user information for usage of the NEDA network as well as lessons in the technology needed to fulfill the goals of NEDA as stated in the charter.

b. This document is delivered annually to each and every paid member of the club. This document should be updated at least once annually to reflect the current state of networking technology in use by NEDA.

c. The Annual is the responsibility of all of the board members. The board may delegate the task of production and mailing of the Annual but maintain the responsibility.

17. Changes to the Constitution

a. Changes to the Constitution may only be made by the following process:

b. At a regularly scheduled quarterly Board of Directors meeting a proposal for a change is submitted in printed or typed form (8 copies) to each of the Directors, to the editor and to the secretary. The item must be presented in person by a NEDA voting member.

c. The format of the submission is in bulleted sections. The following sections must be included: TITLE, PRESENTED, BY, BRIEF, SPECIFICS, PURPOSE. The page is headed with "Constitutional Change Request". TOPIC is followed by one line which identifies the change request. PRESENTED is followed by the date of the board meeting. BY is followed by the name and callsign of the author. BRIEF is followed by a single paragraph description of the change. SPECIFICS is followed by a paragraph by paragraph description of the changes including reference section and paragraph numbers. PURPOSE is followed by a justification for the change. A sample change is available from the club.

d. The proposed change is entered into the minutes of the Board of Directors meeting at which it is presented. Discussion may follow. No vote is taken at this time.

e. At the following board meeting the change is brought up as old business and after discussion is either ratified or not. No change is made if a tie occurs.

f. If a change is ratified then the new copy of the Constitution is printed in the following Quarterly in its entirety.

18. Changes to Bylaws

a. Changes to the bylaws may be made at a single board meeting with the vote of a majority of the board members present. If a tie occurs then no action is taken.

19. Grounds for Dissolution

a. If the board of directors doesn't hold 4 board meetings during the year or if the club is unable to hold elections or there were not three eligible and willing candidates or if the Quarterly in at least it's minimum form) isn't delivered on time then the club must be dissolved.

20. Dissolution of the Club

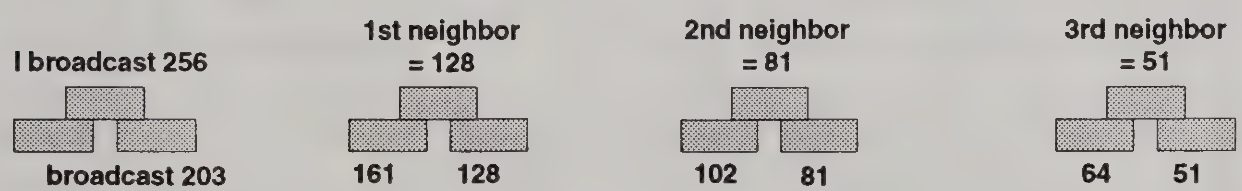
a. After paying out any pending bills the treasurer is directed to write a check for the remainder of the club treasury to the American Cancer Society and to close the all club bank accounts. The name of the club (i.e. North East Digital Association) and it's logo NEDA become the property of the founders of the club, WA2WNI, WA1TPP, KA2DEW, K1MEA, NQ1C, WA2VAM, KC3BQ, to do with as they wish. All paperwork pertaining to software management of individual nodes is delivered to the node/site managers.

TheNET Sysop's Help Sheet

Parameter Function v1.1 & 1.16		LAN[1]	Bkbn	U/G
1	Destination List Length	100	100	100
2	Minimum Quality For Auto Update1	50	50	50
3	HDLC Channel Quality	0	203	50
4	RS-232 Channel Quality	203	203	203
5	Obsolescence Count Init Value	3	3	3
6	Obsolescence Count Min For Broadcast	4	1	4
7	Nodes Broadcast Interval (sec)	1800	1800	1800
8	Time-to-live Initializer (hops)	7	1	7
9	Transport Timeout (sec)	200	200	200
10	Transport RETRIES	2	2	2
11	Transport ACK Delay (sec)	1	1	1
12	Transport Busy Delay (sec)	180	180	180
13	Transport Window Size	2	2	2
14	Congestion Control Threshold	4	4	4
15	No-Activity Timeout (sec)	7200	300	7200
16	P-persistance (see text)	128	255	64
17	Slot Time (10ms)	20	1	20
18	FRACK (sec)	4	1	9
19	MAXframe	1	1	1
20	Link RETRIES	10	10	10
21	Link RESPTIME [t2 timeout] (10ms)	50	20	50
22	Link T3 Timeout [CHECK] (10ms)	65000	65000	65000
23	Digipeating 0=no; 1=yes	0	0	0
24	Validate Callsigns 0=no; 1=yes	1	1	0
25	Station ID 0=msgs;1=after; 2=always	1	0	1
26	CQ Broadcasts 0=no; 1=yes	1	0	1

This node:	256, 203
2nd node:	161, 128
3rd node:	102, 81
4th node:	64, 51
5th node:	40, 32
6th node:	25, 20
7th node:	16, 13
8th node:	10, 8

Parameter Function v2.08		LAN	Bkbn	U/G
1	Minimum Quality For Auto Update1	50	50	50
2	HDLC Channel Quality	0	203	50
3	RS-232 Channel Quality	203	203	203
4	Obsolescence Count Init Value	3	3	3
5	Obsolescence Count Min For Broadcast	4	1	4
6	Nodes Broadcast Interval (sec)	1800	1800	1800
7	FRACK (sec)	4	1	9
8	MAXframe	1	1	1
9	Link RETRIES	10	10	10
10	Digipeating 0=no; 1=yes	0	0	0
11	Validate Callsigns 0=no; 1=yes	1	1	0
12	Host Mode Connects	0	0	0
13	TxDELAY (10ms)	35	35	35
14	Broadcast Via Port b0=radio; b1=RS-232	2	3	3
15	Pound Node Propagate 0=no; 1=yes	0	0	0
16	Connect Command Enable 0=no; 1=yes	1	0	1
EPROM parameters				
17	Destination List Length	100	100	100
18	Time-to-live Initializer (hops)	7	1	7
19	Transport Timeout (sec)	200	200	200
20	Transport RETRIES	2	2	2
21	Transport ACK Delay (sec)	1	1	1
22	Transport Busy Delay (sec)	180	180	180
23	Transport Window Size	2	2	2
24	Congestion Control Threshold	4	4	4
25	No-Activity Timeout (sec)	7200	300	7200
26	P-persistance (see text)	128	255	64
27	Slot Time (10ms)	20	1	20
28	Link RESPTIME [t2 timeout] (10ms)	50	20	50
29	Link T3 Timeout [CHECK] (10ms)	65000	65000	65000
30	Station ID 0=msgs;1=after; 2=always	1	0	1
31	CQ Broadcasts 0=no; 1=yes	1	0	1
32	Heard List Length	20	20	20
33	Full Duplex 0=no; 1=yes	0	0	0



This drawing represents the node quality value for a single node as it propagates through several node hops.

Notes:

Note changes to the HDLC Channel Quality value. It was incorrectly listed as 202 in the last Quarterly.

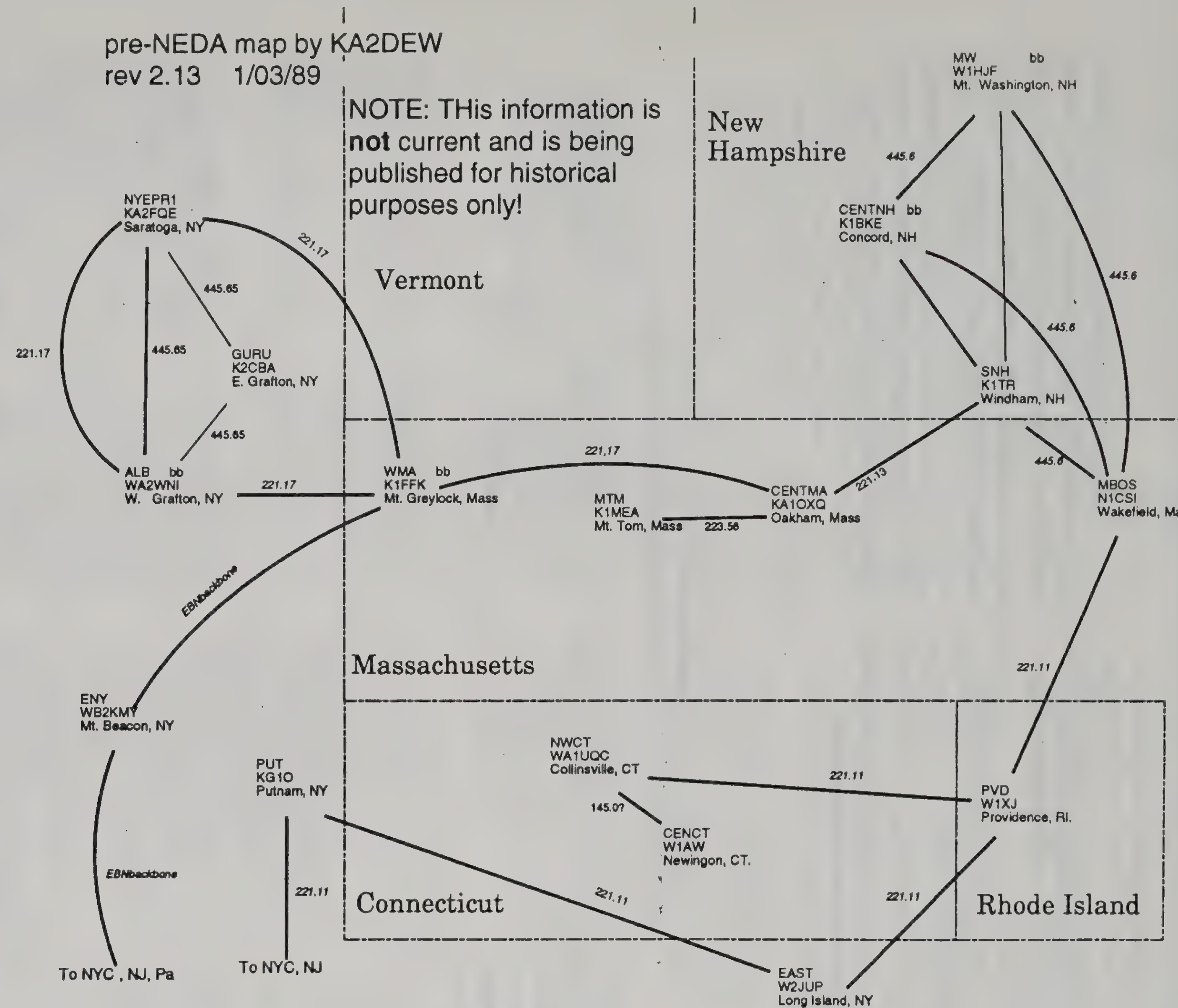
A LAN port is a port that is on a frequency which has no other nodes nor any servers. The port would be used by stations who mostly aquire data from the network. These parms would be incompatible with a crowded channel.

U/G indicates a port on a frequency which would be used by users and/or servers and/or other nodes. This includes LAN frequencies which have, since creation, have aquired KAnodes, digipeaters and/or other nodes and servers.

Bkbn indicates a port that talks to a single other node which is similarly configured.

—Editor

BBScall	nodename:nodecall
KA1MGO-1	MBOS2 :N1CSI-2
WB1DSW-1	MBOS2 :N1CSI-2
N1BGG	MBOS2 :N1CSI-2
WA2PVV	WMA :K1FFK
WA2WNI-4	ALB2 :WA2WNI-2
K1MEA-4	MTM :K1MEA-1
WA2UMX	#WMA2 :K1FFK-12
N1API	MTM :K1MEA-1
KB4N-4	SNHUHF:K1TR-2
WA2YVL	WMU :W1HJF-4
WA2RKN-2	ENY :WB2KMY-1
K1UGM	SNHUHF:K1TR-2



USER PORT INFO:

site	freq.	mode	callsign
ALB	144.93 -	ALB144	:WA2WNI-1
	223.58 -	ALB220	:WA2WNI-2
ENY	145.59 -	ENYLAN	:WB2KMY-2
	145.01 -	ENY	:WB2KMY-1
WMA	145.05 -	WMA	:K1FFK
MTM	145.09 -	MTM	:K1MEA-1
NYEPR1	145.01 -	NYEPR1	:KA2FQE
SNH	145.07 -	SNH	:K1TR-1
	441.025-	SNHUHF	:K1TR-2
CENTNH	145.05 -	CENTNH	:K1BKE
MW	145.01 -	MWV	:W1HJF-1
	445.60 -	MWU	:W1HJF-4
CENTMA	145.07 -	CENTMA	:KA1OXQ-1
MBOS	145.01 -	MBOS1	:K1UGM-1
	221.11 -	MBOS2	:K1UGM-2
	445.60 -	MBOS4	:K1UGM-4

Networking Rules:
1> No hidden transmitters on backbones.
2> No nodes of poor reliability pass into backbone.
3> No single port nodes.
4> Redundancy is more important than added coverage.

bb = emergency power   *Frequency* = Backbone with HTS

NEDA HexiPus™ Order Form

Use this order form when purchasing HexiPus™ board kits by mail from the POBox or from a NEDA consignee at a special event.

Use the latest version of this form if possible. See bottom of the page for release date.

You do not have to pay shipping if you are getting the HexiPus™ from a NEDA agent/consignee.

To Consignee: Please make sure that each purchase is handled with one of these forms. Correctly document funds exchanged, check numbers, purchaser's name and address if not by cash; and quantities of each kit type delivered.

To Mail Purchaser: Please fill out all sections of the form except those marked "For Office Use Only".

This will help our treasurer track the sales of the product so that our club may be run efficiently and above board.

Thank you and good luck with your node!

Purchaser Information

Name

Address

City State/Province

Country Zip

Callsign (Optional) Date Purchased

✔Cash Check Number ✔US bank ✔Canadian

Amount ↓

If funds are Canadian compute exchange rate as best you can. If check is drawn on a Canadian bank add $5 U.S. to total.

# of Board+Diode Kits (US) x$22.95 + # of Complete Kits (US) x$29.95

subtotal in US funds:

If by mail add $4.00 per unit
No mail delivery outside U.S.

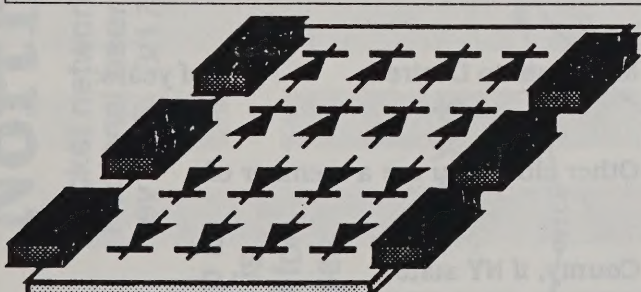
Total in US funds:

Agent/Consignee Information

Name

Event

Notes: (specify connector type, male/female)



The NEDA HexiPus™ Kit is, as of this printing, available in three formats. Board with diodes; Board with diodes and female connectors; or board with diodes and male connectors. The supply of the female connectors is limited as they were purchased surplus. Please specify your preference in the notes section above. The shipper will give you male connectors if females run out.

Funds for the sale of HexiPus™ boards go to NEDA. NEDA is a non-profit club.

For Office Use Only (Treasurer)

Date Rx by Treasurer: **Date Tx to HexiPus™ Cmty** **Deposit**

Order Form Date: 2/1/92

North East Digital Association Membership Application

Welcome to packet networking. This is the official membership application form for N.E.D.A.

Some General Stuff About N.E.D.A.:

N.E.D.A. was founded on Sept. 17, 1989. N.E.D.A. holds 4 scheduled yearly board meetings announced one month in advance. During each year the meetings are held in different cities in the areas most saturated by the membership. The club committees sponsor technical sessions throughout the year in various locations as well. The board meetings are open to voting members only, all of whom are invited via packet mail @ their Home BBS. Technical meetings are open to all and are held when scheduled at sites selected by the membership.

Club funds may only be allocated at the board meetings, the minutes of which are printed in the *Quarterly*.

The board of directors consists of 6 hams who are elected for 2 year terms by the voting membership. The board of directors appoints an editor, membership director, treasurer plus any additional department heads as they see fit.

The dues structure of NEDA is as follows:

Associate network support membership is $10. Associate network support membership with quarterly updates is $15. Voting membership in N.E.D.A. is $25. Voting members decide which 3 members will be appointed to the board of directors at the general meeting.

Canadian membership applications should include U.P.S. money orders or other bank draft which can be converted to U.S. funds at low cost. Checks drawn off of Canadian banks should include $15 additional in U.S. funds to cover processing fees. Multiple years of dues or dues for multiple applications only need include the additional funds for one check. Canadian memberships should include an additional surcharge of $5 per year per membership to cover postage.

Non North American applicants should contact the club at the POBox before sending funds.

Membership in the U.S. is $15/year for Associate Membership with Quarterly or $25/year for Voting.
Membership in Canada is $20US/year for Associate Membership with Quarterly or $30US/year for Voting.
Mail completed application with check to: **NEDA POBox 563 Manchester NH 03105**.
$10 of the NEDA membership dues, for the year starting at date of receipt, is for a subscription to the NEDA Quarterly for one year. Return this bill form with remittance.
If paying with check drawn from a Canadian bank please add $15U.S. Please compute exchange rate and make amount equal the dues amount in U.S. funds.

Name:

Home Phone Number:

Address:

Work Phone Number:

City: State:

FIDO, Bitnet, Compuserve or Internet address:

Country: Zip:

Membership Desired: # of years:

Callsign:

☐ Check if this form is a renewal or information update.

Other clubs you are a member of:

Home BBS (include hierarchal address)

County, if NY state

The BBS address is used for club mailings so please keep the club informed of any changes.

☐ Check if RACES or ARES member

FOR OFFICE USE ONLY:

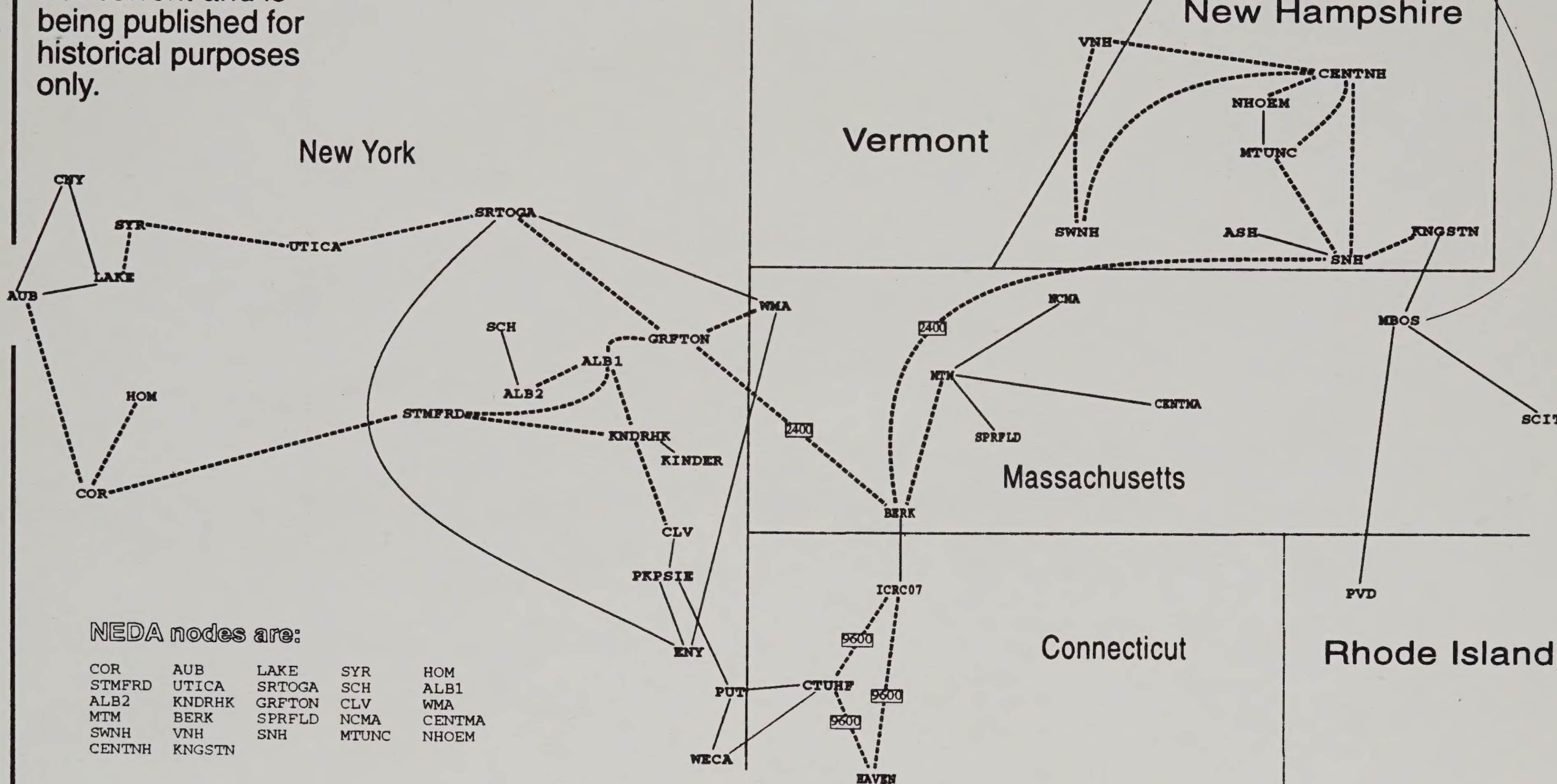
*RCT:*________ *CNO:* ________ *ACK:* ________ *DDP:* ________ *DOE:* ________ *PKG:* ________

North East Digital Association

packet network
Regional **user port** map
rev 1.05   3/17/90

Send SASE for membership application and info to:
NEDA
Box 563
Manchester NH
03105

NOTE: This map is **not** current and is being published for historical purposes only.



NEDA nodes are:

COR	AUB	LAKE	SYR	HOM
STMFRD	UTICA	SRTOGA	SCH	ALB1
ALB2	KNDRHK	GRFTON	CLV	WMA
MTM	BERK	SPRFLD	NCMA	CENTMA
SWNH	VNH	SNH	MTUNC	NHOEM
CENTNH	KNGSTN			

NEDA - Archive map. Printed in Quarterly #2.4
Not Current!!

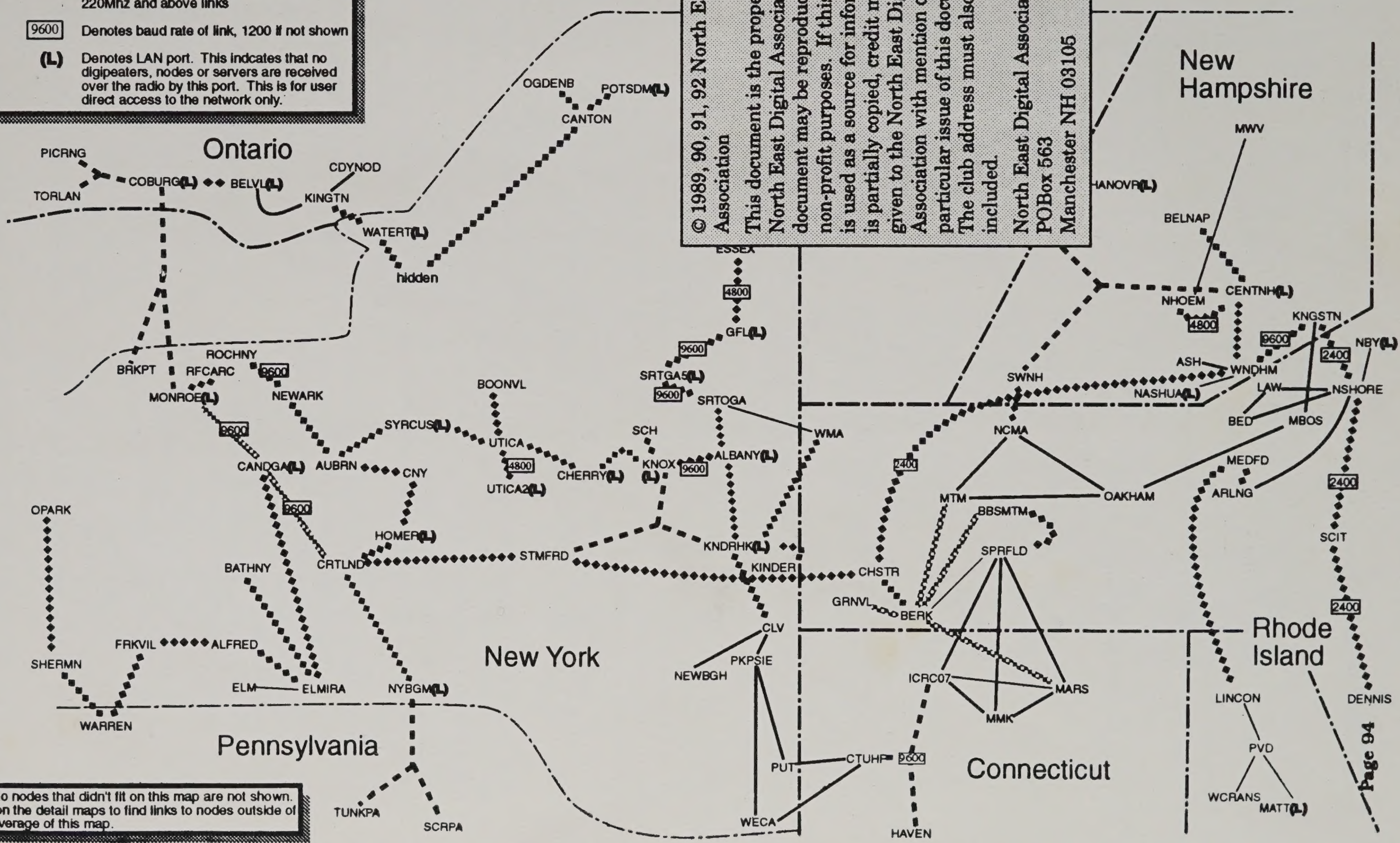
SE ONT, NY, MA, VT, NH, RI, CT Abbreviated map
1/15/92 v2.13

For TheNET, G8BPQ, NOS, MSYS and NET/ROM, the N command at some nodes shown may only show the immediate neighbor across some links. You must single step across those links.
This map shows all known general purpose packet radio nodes that are interconnected via hidden transmitter free amateur radio backbones in the area of the map. This map also shows all general purpose nodes within one hop of the above mentioned nodes on 50MHz or 220MHz and up.

- ◆◆◆◆◆◆ Dedicated Point to Point Backbone
- ■ ■ ■ ■ Multi-Way HTS Free Backbone
- oooooooo Via Repeater
- —— Non HTS Free or not sure 50MHz or 220Mhz and above links
- 9600 Denotes baud rate of link, 1200 if not shown
- (L) Denotes LAN port. This indcates that no digipeaters, nodes or servers are received over the radio by this port. This is for user direct access to the network only.

Membership info for the North East Digital Association is available for a SASE to:
NEDA
Box 563
Manchester NH 03105

North East Digital Association
POBox 563
Manchester NH 03105



Links to nodes that didn't fit on this map are not shown. Look on the detail maps to find links to nodes outside of the coverage of this map.